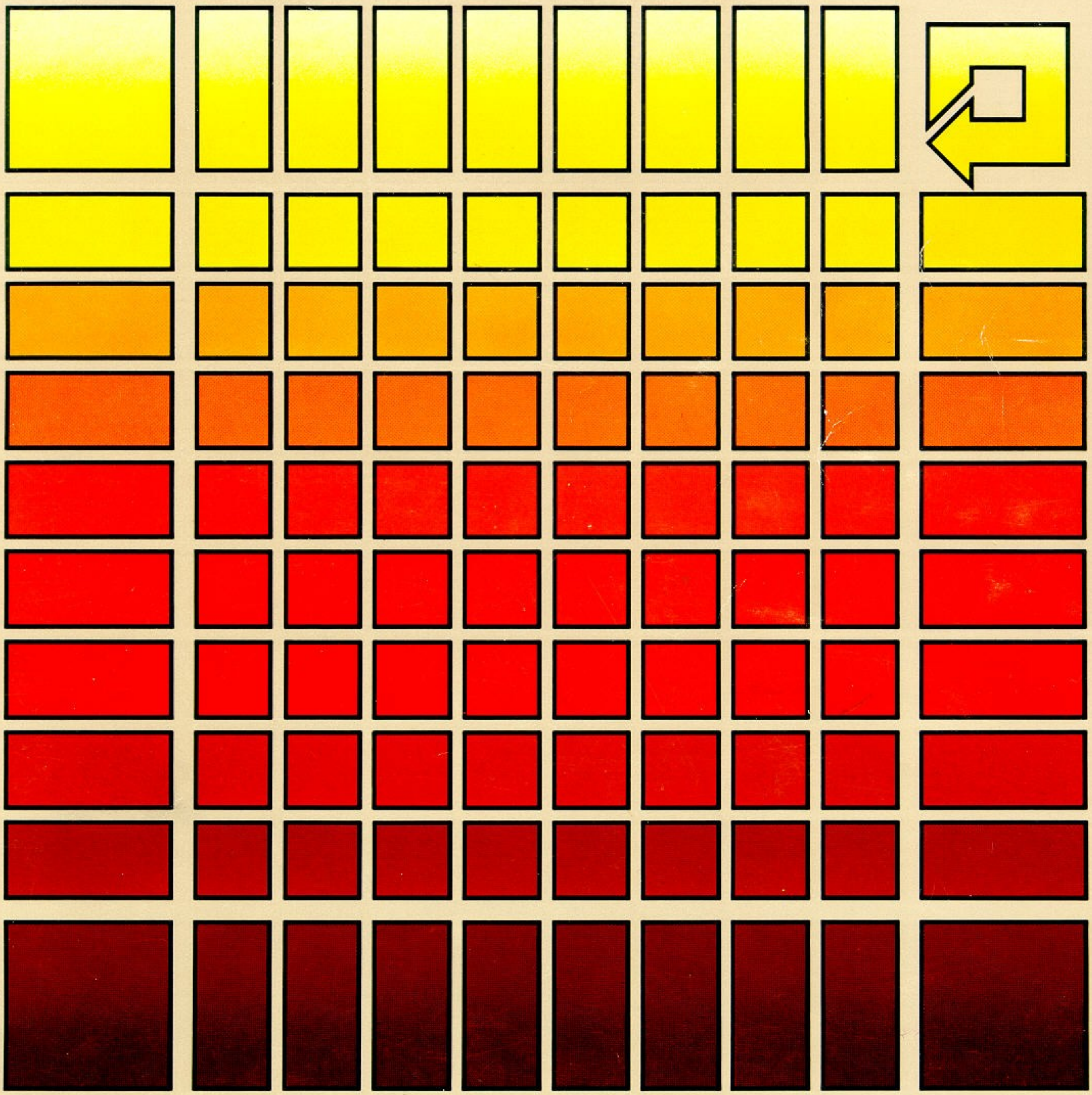
HEWLETT-PACKARD

HP 82164A

HP-IL/RS-232-C Interface

OWNER'S MANUAL



HEWLETT PACKARD

HP 82164A
HP-IL/RS-232-C Interface

Owner's Manual

March 1983

82164-90002

Printed in U.S.A.

Contents


82164A HP-IL/RS-232C INTERFACE
HEWLETT-PACKARD
PWR
T/R
RESET

Read This First!

System Documentation

This manual for the HP 82164A HP-IL/RS-232-C Interface must be used in conjunction with the manuals for other components in your HP-IL/RS-232 system. In particular, the manuals for your HP-IL controller and its HP-IL enhancements will describe how you'll control the interface and the RS-232 device connected to it. It is important to recognize that the main content in this manual describes the capabilities of only the interface, and that you'll need to refer to other manuals for additional information. For example, you'll need to refer to the manual for the RS-232 device to determine its operating characteristics.

To best learn how to use the interface and RS-232 device with your controller, read these parts of this manual in the order shown:

1. **Section 1, "General Information."** It gives basic information about connecting the interface and RS-232 device and gives an overview ot the interface's operation.

   Note: If you aren't familiar with how your controller performs HP-IL operations, look through the appendix that corresponds to your controller. This shows how you'll control the interface and RS-232 device. (The appendix refers to features of the interface that are described in the body of this manual, so you may not understand some details at this time.)

2. **Sections 2, 3, and 4.** They present technical information about the interface's operation. (Refer to appendix D as required for reference information about the control and character registers.)
3. **Sections 5, 6, and 7.** They give specific information about using the interface with certain classes of RS-232 devices. These sections show connections for specific devices as an aid for connecting your particular system.
4. **The appendix that corresponds to your controller.** It lists the interface's capabilities that your controller can use, describes how your controller operates the interface and RS-232 device, and gives several examples that show typical operation.

If you have a controller that isn't discussed in an appendix, refer to section 3 for information about how the interface operates using HP-IL in general. If the controller is manufactured by Hewlett-Packard and you need information about using it with the interface, you can write to:

Hewlett-Packard Company
Corvallis Division Customer Support
1000 N.E. Circle Blvd.
Corvallis, OR 97330

General Information

Introduction

The HP 82164A HP-IL/RS-232-C Interface provides the capability to interface an external device having serial input/output (RS-232-C) capabilities with the Hewlett-Packard Interface Loop (HP-IL). (This manual uses "RS-232" as an abbreviation for RS-232-C.)

The HP 82164A HP-IL/RS-232-C Interface is packaged with the following accessories:

- One HP-IL cable.
- An ac adapter.

Additional HP-IL cables are available individually in the following lengths:

- ½ meter (1½ feet)—model number HP 82167A.
- 1 meter (3 feet)—model number HP 82167B.
- 5 meters (16 feet)—model number HP 82167D. (This length may not be available in all countries.)

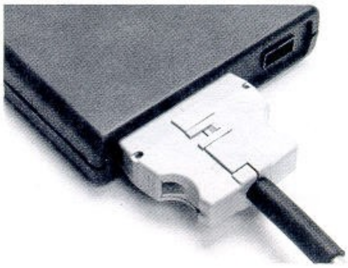
This manual gives information about the interface's design, its interaction with HP-IL, and its operation using the RS-232 capabilities of the external device. Operating specifications are included to assist in properly connecting the interface and the external device.

Installation

The following paragraphs describe how to set up the HP 82164A HP-IL/RS-232-C Interface in an HP-IL system.

RS-232 Connection

To connect an external device to the interface, first wire the device to a female 25-pin D-subminiature RS-232 connector.* (Refer to section 2 and appendix B for information about RS-232 signals.) Then disconnect the power from the interface and plug the connector into the interface's RS-232 receptacle.



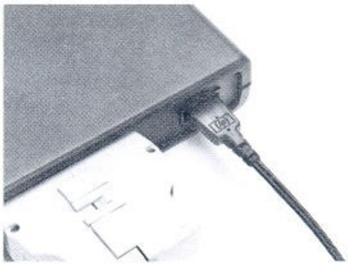
Power

The HP-IL/RS-232 interface is powered by an ac adapter. Because the interface does not have a battery, it can operate only when the adapter is connected to the interface and a proper ac outlet. To install the adapter,

* The interface is less susceptible to interruption by electrostatic discharge (ESD) if you use a nonmetallic connector.

first make sure that the interface is disconnected from HP-IL. Then, insert the ac adapter plug into the proper ac outlet and insert the power connector into the power receptacle in the interface's rear panel.

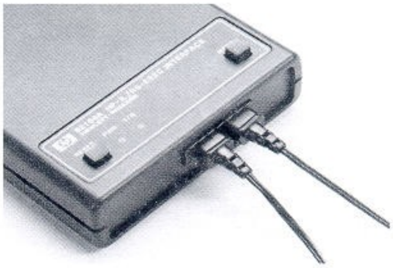
Finally, press the RESET key to initialize the interface to its startup conditions.



HP-IL Connection

The Hewlett-Packard Interface Loop (HP-IL) consists of one or more peripheral devices and a controller (such as a computer). The devices may be connected in any order—but all of the interface cables must form a continuous loop. All connections are designed to ensure proper orientation.

To connect the HP 82164A HP-IL/RS-232-C Interface to the loop, first turn off the controller. Then disconnect the loop in one place and connect the interface into the loop at that place. (In some instances, the interface may be the only peripheral in the loop.)



Keyboard

The keyboard on top of the interface contains a few controls that enable you to set and monitor the interface's operation.

RESET Key. The RESET key is a momentary switch that returns the interface to its startup conditions. (Refer to page 10.)

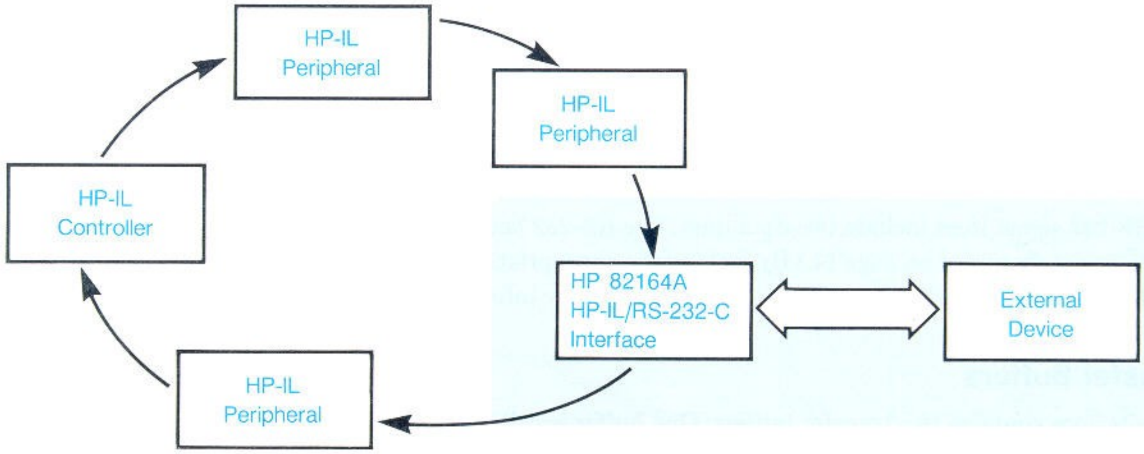
PWR Light. The PWR (*power*) light is on whenever the ac adapter supplies power to the interface.

T/R Light. The T/R (*transmit/receive*) light turns on each time data is transferred across the RS-232 data lines. The light is lit only for the duration of the transfer, so it will often appear to flash or blink.

MSRQ Key. Pressing the MSRQ (*manual service request*) key sets a manual service request condition in the status register. This condition remains after the key is released. (Refer to page 43 for more detail.)

An Overview of the Interface's Operation

Consider the HP-IL system shown below. The interface loop contains an HP-IL *controller* (such as a computer), perhaps one or more additional HP-IL devices, and the HP-IL/RS-232-C interface. The interface connects to an *external device* (such as an RS-232 serial printer), allowing the controller to interact indirectly with the external device. In this way, the external device becomes an HP-IL controlled peripheral.



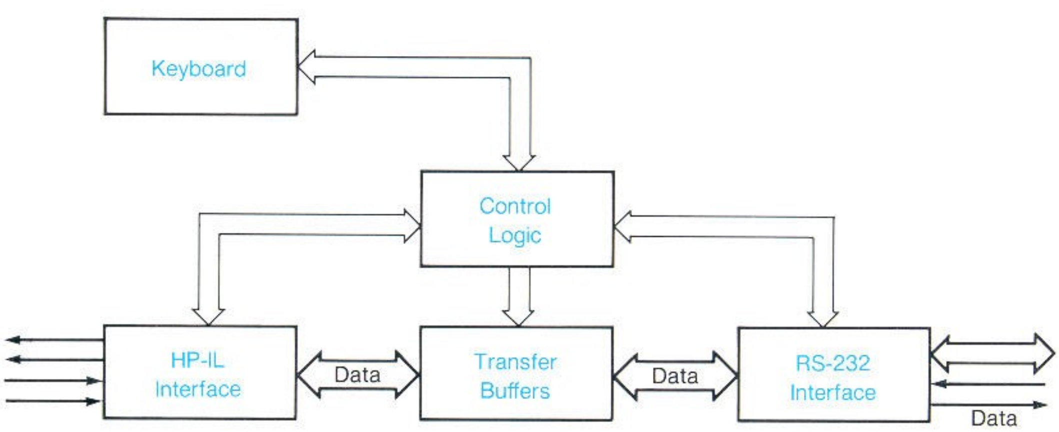
If the controller needs to send data to the external device, the controller first makes the interface a listener, which means that the interface is set to accept data from HP-IL and pass it to the external device. The controller then initiates the transfer of data around the interface loop, one character (or byte) at a time. As characters are received by the interface, it stores them internally. Meanwhile, the interface sends the data to the external device one character at a time.

If the controller needs the external device to send data to listeners on HP-IL, the controller first makes the interface a talker, which means that the interface is set to accept data from the external device and send it on HP-IL. The controller then directs the interface to start sending data.

This example illustrates one way that the interface can be used. However, it can be set up to operate in several different ways. Using the interface with HP-IL and an external device requires an understanding of these options. Read through this manual in its entirety before attempting to connect and use the HP 82164A HP-IL/RS-232-C Interface with a particular external device.

Internal Design

The HP-IL/RS-232 interface has five primary features that are important for understanding the interface's operation: the HP-IL interface, the RS-232 interface, the transfer buffers, the control logic, and the keyboard.



HP-IL Interface

The HP-IL interface portion of the interface performs standard operations required by the interface loop, such as maintaining the interface's talker or listener status, and accepting and passing HP-IL messages around the loop. The physical connection to HP-IL consists of standard HP-IL receptacles—one for incoming messages and one for outgoing messages.

RS-232 Interface

The RS-232 interface portion of the interface provides the connection to the external device. The physical connection consists of a male 25-pin D-subminiature RS-232 receptacle. An internal *configuration selector* enables you to select one of two possible pin configurations for the RS-232 receptacle. ("Changing the Configuration" on page 21 contains more information about the selector.)

The RS-232 signal lines include two data lines, five RS-232 handshake lines, and a ground line. (Signal descriptions are discussed on page 14.) By making the appropriate connections, you enable the external device and the interface to use the signal lines to send and receive information.

Transfer Buffers

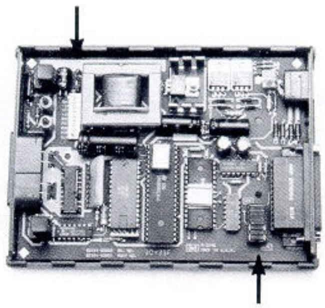
The interface contains two transfer buffers. One buffer is called the transmit buffer and contains the data being transmitted to the RS-232 device. The other buffer is called the receive buffer and contains the data received from the RS-232 device.

The transmit buffer is capable of holding 84 bytes, and the receive buffer is capable of holding 109 bytes. (Each byte consists of eight bits.) The buffers pass data in the order it is received—first in, first out.

Control Logic

The control logic stores operating information, implements various operating modes that can be selected, and controls the flow and interpretation of data within the interface. It includes registers that store operating information: the control registers, the character registers, and the status registers. This operating information can come from the HP-IL controller or the interface's keyboard.

An internal switch enables you to control the service request conditions that are active at startup or reset. (Refer to "Service Requests" on page 43 for more information.)



Keyboard

The keyboard contains the RESET key, the MSRQ key, the PWR indicator light, and the T/R indicator light. The RESET key and the MSRQ key allow you to interact with the interface and set certain states. (The operation of these two keys are explained on page 8). When the PWR indicator light is lit, there is power to the interface. The T/R light is lit during the transfer of data and during other activity.

Startup Conditions

Each time you connect the ac adapter to the interface or press the RESET key, the interface performs a self-test, clears its transmit and receive buffers, and initializes itself according to the default parameters in the control and character registers. (Refer to appendix B.) The interface's HP-IL address is undefined, preventing the interface from performing any HP-IL operation until it is assigned a valid address by the HP-IL controller.

The self-test mentioned above checks part of the interface's circuitry. During the test, the PWR and T/R lights are turned on to verify their operation. They both remain lit for about 2 seconds. If the T/R light stays lit after the self-test, the interface has failed the self-test and requires repair—it will not operate in this condition.

Section 2

RS-232 Operation

This section presents some of the more technical operating aspects of the RS-232 side of the HP 82164A HP-IL/RS-232-C Interface. You may not need some of the material presented in this section for your specific application. You may want to read only those parts of this section that apply to your application.

The following list contains the various options that are frequently found on RS-232 devices and that are implemented on your interface. Also listed are the page references where you can find additional information on how the interface implements these options.

- Pin configurations (pages 14 and 21).
- Full-duplex and half-duplex operation (pages 15 and 16).
- Hardware handshakes (page 17).
- Software handshakes, including XON/XOFF and ENQ/ACK (page 18).
- Block size for buffer control (page 19).
- Baud rate (page 20).
- Parity options (page 20).
- Word length options (page 20).
- Stop bit options (page 20).
- Echoing back to the external device (page 34).
- End-of-line character detecting and deleting or inserting (page 21).
- Waiting (by sending nulls) after inserting an end-of-line character (page 81).

Most options are selected by certain bits in the control registers. In this manual, individual bits are indicated by appending the bit numbers to the register name. For example, bits 2 and 1 of control register R02 are indicated by R02-2,1. Control registers are discussed in detail in section 4 and appendix D.

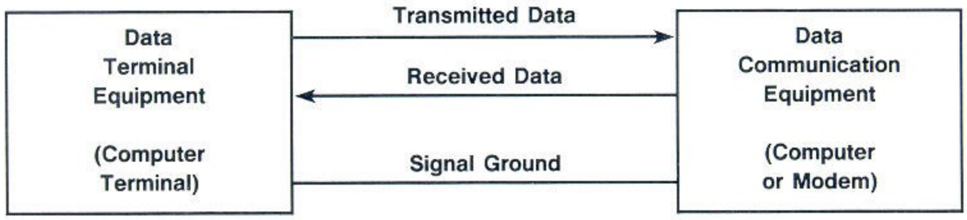
RS-232 Description

RS-232 is a serial input/output (I/O) interfacing method that transmits data one bit at a time over one transmission wire. For example, an eight-bit character would take eight sequential transmissions to complete the character. In contrast, parallel I/O transmits each bit simultaneously over several wires. Sending an eight-bit character would take eight separate wires, one wire for each bit, and all eight bits would be sent simultaneously. One disadvantage of serial I/O is that it is much slower than parallel I/O. But the major advantage of serial I/O is that it is much less expensive than parallel I/O, especially over long distances.

RS-232 can be a very difficult interface to work with because RS-232-C is only *partially* specified by the Electronic Industries Association Standard of 1969. Additionally, RS-232 was purposefully made general enough to include most of the serial interfaces that were developed prior to the 1969 standard. Thus, you may frequently find an RS-232 device that does not work with another RS-232 device without some sort of modification. Your HP-IL/RS-232 interface has been structured so that most modifications can easily be made.

Equipment Configurations

Serial I/O devices are described according to the function that they perform. These functions are Data Terminal Equipment (DTE), Data Communication Equipment (DCE), and current loop operation. Current loop operation is not supported by the HP 82164A HP-IL/RS-232-C Interface.



A Data Terminal Equipment (DTE) is a device at any location in a network where information can enter or exit. Examples of DTE are:

- Remote terminals.
- Remote terminal interfaces.
- Host computer.
- Host computer interface.

A Data Communication Equipment (DCE) is a device used to convey information between locations. Examples of DCE are:

- Modems.
- Modem interfaces.
- Communication links (telephone lines, for example).

According to the RS-232 standard, Data Terminal Equipment should have a male connector and Data Communication Equipment should have a female connector.

Note that these standards are not adhered to by all equipment manufacturers, so some devices, such as serial printers, are configured as either DTE or DCE, depending on the manufacturer.

Signal Lines

The RS-232 signal lines supported in the HP-IL/RS-232 interface are listed in the table below. Also listed are the pin assignments for the interface in its standard configuration as a Data Terminal Equipment. (Refer to page 21 for optional configurations.)

Pin	Signal	Signal Direction
2	Transmitted Data	HP 82164A → external device
3	Received Data	HP 82164A ← external device
4	Request To Send	HP 82164A → external device
5	Clear To Send	HP 82164A ← external device
6	Data Set Ready	HP 82164A ← external device
7	Signal Ground	HP 82164A ↔ external device
8	Received Line Signal Detect	HP 82164A ← external device
20	Data Terminal Ready	HP 82164A → external device

All signal lines are defined from the view of the Data Terminal Equipment. For example, the Data Terminal Equipment sends data on the Transmitted Data line and receives data on the Received Data line. However, the Data Communication Equipment receives data on the Transmitted Data line and sends data on the Received Data line. In determining how your external device is configured, it is very important to determine if

the external device sends data on pin 2 or 3. If the device sends data on pin 2, then it is configured as a DTE. If the device sends data on pin 3, then it is configured as a DCE. Your HP 82164A HP-IL/RS-232-C Interface in its standard configuration is a DTE.

Note that some serial printers may have a male (DTE) connector, and yet the pin locations are for a DCE (female) connector.

The signals listed above are defined in logical groups—not according to pin assignments—in the following paragraphs.

Signal Name	Definition
Transmitted Data (pin 2)	DTE's output data line and DCE's input data line.
Received Data (pin 3)	DTE's input data line and DCE's output data line.
Signal Ground (pin 7)	Establishes signal reference level between DTE and DCE.

There are three modes of RS-232 operation: full-duplex, half-duplex, and simplex. Full-duplex means that the equipment can send and receive information simultaneously. For example, a DTE could be sending data on the Transmitted Data line and also be receiving data on its Received Data line. Half-duplex means that the equipment can both send and receive data, but not at the same time. For example, a DTE could send data on its Transmitted Data line and, after it was finished, it could receive data on its Received Data line. Simplex means that the equipment can only send or only receive data, but not both. A simplex device would have only a Transmitted Data or Received Data line. The HP 82164A HP-IL/RS-232-C Interface always operates as a full-duplex device, but it can operate with a half-duplex external device.

Signal Name	Definition
Request To Send (pin 4)	Notifies DCE that DTE is ready to send data.
Clear To Send (pin 5)	Notifies DTE that DCE is ready tc accept data.
Data Set Ready (pin 6)	Notifies DTE that DCE has power on and is not in test or dial mode.
Received Line Signal Detect (pin 8)	Notifies DTE that DCE is receiving carrier signal.
Data Terminal Ready (pin 20)	Notifies DCE that DTE is ready to transfer data. (Inhibits DCE from sending when false.)

These five lines perform coordinating functions known as handshakes. A handshake is simply a method for one device to tell another device its status or what it is ready to do. For example, the Data Terminal Equipment could set the Data Terminal Ready line true to tell the Data Communication Equipment that the Data Terminal Equipment is ready to either send or receive data. The Data Communication Equipment will then set the Data Set Ready line true to tell the terminal that it is ready. These are hardware handshake lines. They are used for channel control, not for controlling the passing of information. These lines indicate the status of the hardware—whether it is on or off and if it is functional.

The Received Line Signal Detect line may or may not be used by the RS-232 device. This signal is set true by a modem whenever it is receiving a good carrier signal. In this way the interface knows it is able to send or receive information through the modem. Additionally, some printers use this name for an input signal that must be true before they will print.

Full-Duplex Operation

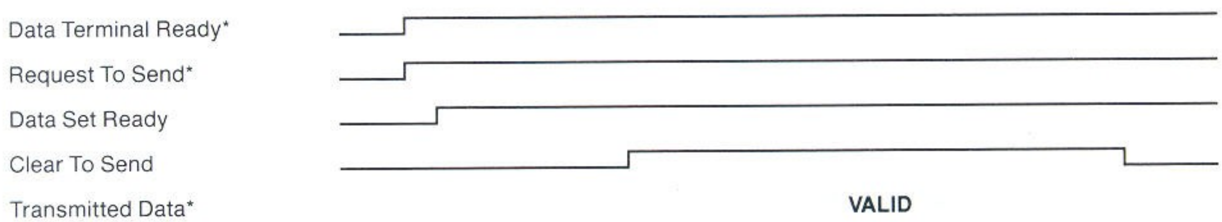
The HP-IL/RS-232 interface is a full-duplex device—it can send data and receive data simultaneously. The following paragraphs describe how the interface would use the hardware handshake lines while it is connected to a full-duplex external device. (Many devices do not require a hardware handshake—refer to "Hardware Handshake Options" on page 17.)

For full-duplex operation, the interface monitors the Data Set Ready input line to determine if the external device is ready. The Clear To Send input line must also be true before the interface will send data to the external device on the Transmitted Data line; the Received Line Signal Detect input line must also be true before the interface will accept data on the Received Data line.

The interface sets and keeps its Data Terminal Ready and Request To Send output lines true as soon as the interface is ready to operate. (It sets and keeps Data Terminal Ready false if it initiates an autodisconnect condition—refer to page 45.)

For example, the following diagrams illustrate possible sequences of signal conditions for the transfer of data. (Signals marked with * are sent by the interface.)

Sending Data:



Receiving Data:

Certain RS-232 devices may requires that the signal lines be set true in a certain order; other devices may set all signal lines true when they are turned on. The important point to observe from this example is that the interface requires that certain signals be true before it is ready to send or receive data on RS-232.

Operating With Half-Duplex Devices

The HP-IL/RS-232 interface can be used with a half-duplex device. Even though the interface will be able to perform full-duplex communication, its hardware handshake can properly control the half-duplex device.

For half-duplex operation, the interface monitors the Data Set Ready input line to determine if the external device is ready. The Received Line Signal Detect input line must also be true before the interface will accept data on the Received Data line.

The interface sets and keeps its Data Terminal Ready output line true as soon as the interface is ready to operate. (It sets and keeps Data Terminal Ready false if it initiates an autodisconnect condition—refer to page 45.)

Normally, for half-duplex operation, the interface should use the auto Request To Send option (enabled by R10-0 equal to "1"). With this option enabled, the interface uses its Request To Send output line to indicate the condition of its transmit buffer (the output is false if the buffer is empty, true if the buffer isn't empty) except to indicate the end of each line of data (the output goes false at the end of each line). (If the auto Request To Send option were disabled, this output signal would not automatically go false at the end of each line.)

The Request To Send output line and the Clear To Send input line control the direction of data transfer. If both lines are true, the interface sends data from its transmit buffer on the Transmitted Data line. If either line is false, the interface doesn't send data but does accept data from the Received Data line (if Received Line Signal Detect is true), placing the data in its receive buffer.

If the interface has data in its transmit buffer, it tries to set the Request To Send line true—but it doesn't set the line true until the Clear To Send line is false. Subsequently, when the Clear To Send line goes true, the

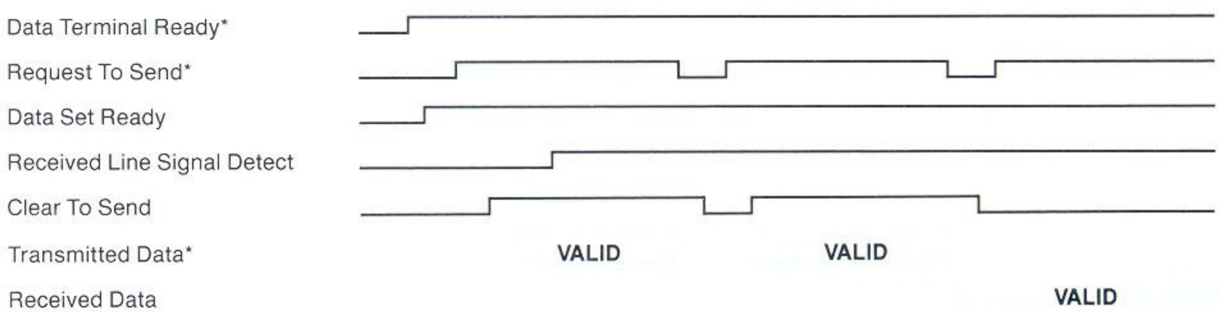
interface can send its data. (This means that the external device must control the Clear To Send line in response to the Request To Send line. If the device doesn't respond properly, you can restore the interface's operation by using the "LE0" or "LE4" Remote instruction or by sending a Device Clear or Selected Device Clear message—refer to sections 3 and 4.)

Any one of three conditions causes the interface to stop sending data: it empties its transmit buffer (Request To Send goes false), it sends an end-of-line indicator (Request To Send goes false), or the external device interrupts the transfer (sets Clear To Send false). Whenever the interface is not sending data (Request To Send false or Clear To Send false), the external device can send data to the interface (if Received Line Signal Detect is also true).

The end-of-line indicator mentioned above provides pauses in the interface's output so that the external device can set its RS-232 channel for transfer to the interface, if necessary. The external device must monitor the Request To Send line and set its Clear To Send line false when Request To Send goes false. However, when Request To Send goes true, the external device may hold Clear To Send false until after it sends a line of data. This prevents the interface from sending data while the external device is sending data. (Clear To Send shouldn't go true unless Request To Send is true.)

The end-of-line indicator is specified in character registers C02 and C03. (Refer to page 83.) The interface recognizes the specified indicator, even though R10 may have disabled the end-of-line delete and insert options.

For example, this diagram illustrates how the hardware handshake lines might control the transfer of data for operation with a half-duplex device. (Signals marked with * are sent by the interface.)



Hardware Handshake Options

The previous topics illustrate how the interface interacts with the external device using the hardware handshake lines. In those discussions, all of the handshake lines are used—the interface is using a "full" handshake. These handshake lines indicate whether the interface and the external device are prepared to send or receive data—they indicate the condition of the communication channel, or hardware.

However, in its default condition, the interface uses *none* of the hardware handshake lines. That is, the interface ignores the input signals and assumes that the external device is ready to send and receive data—the interface doesn't wait for any of the input signals to go true. The interface also keeps its two output handshake lines true (except for automatic disconnect). This is called the "three-wire" connection (referring to the two data lines and the ground line)—it often uses a software handshake to control the data transfer, as discussed in the next topic.

However, you can set the interface to observe any (or all) of the input handshake lines. If a line is observed, the interface uses the line according to the description in the preceding discussions. For example, if the Received Line Signal Detect signal is observed, then this line must be set true by the external device before the interface will accept data on the Received Data line. This enables you to use a hardware handshake that provides optimum communication with the external device. The lines that are to be ignored or observed are specified by control register R09. (Refer to page 80.)

For example, suppose that only Clear To Send is ignored (R09-3,2,1 equal to "001"). Whenever Data Set Ready is true, the interface assumes that the external device is ready to receive data. This situation may apply to certain printers that don't have the capability to indicate data overflow—that is, the printer is always ready for data.

Another hardware handshake option concerns the interface's Request To Send output handshake line. In its default condition, the interface sets and keeps Request To Send true following startup and reset. As discussed above for operating with a half-duplex device, the auto Request To Send option (enabled by R05-3 equal to "1") enables the interface to set this line true and false according to half-duplex conventions. (Refer to page 16.)

The last hardware handshake option enables the HP-IL controller to monitor and control the states of the handshake lines. The controller can monitor the input lines by reading control register R05-2,1,0. It can control the output lines by setting R04-3,2 equal to "11" and then placing the appropriate values in R04-1,0. This feature also enables the controller to keep either output line true or false, while allowing the interface to automatically control the other output line.

Software Handshakes

The handshake lines, in strict RS-232 protocol, are used for controlling the communications channel, not the flow of information. In other words, the hardware handshakes are used to say "My hardware is ready to send or receive information." Because the hardware handshakes indicate the status or condition of only the hardware, not the internal operating status or operating condition of the device or interface, the hardware handshakes are not intended to control the flow of information across RS-232. To temporarily interrupt the flow of information, RS-232 devices frequently use a software handshake.

One common RS-232 configuration is called the "three-wire" connection. Only the Ground, Transmitted Data, and Received Data lines are connected. It is assumed that the hardware is always ready to transfer data. In this configuration, software handshakes are recommended.

Two software handshakes have become industry standards. One software handshake is called the *transmitter protocol*; it typically uses the ASCII characters ENQ and ACK as the control characters. The other software handshake is called the *receiver protocol*; it typically uses the the ASCII characters XON and XOFF as the control characters. (XON and XOFF are frequently referred to as DC1 and DC3.)

More specifically, software handshakes are used to prevent problems that arise because of the increased use of time-sharing on computer networks. If a host computer is temporarily busy, it needs a method to notify terminals on the network that they need to wait. Software handshakes can prevent problems such as data being lost because the host is busy or the buffer is overflowing.

Control register R11 determines the software handshake options. Whenever this register is redefined, the interface is enabled to start sending data to the external device without receiving any additional software handshake characters.

Transmitter Protocol. Transmitter protocol is used by the primary transmitting device to control the transfer of data. The primary device is called the *host* device; the other device is called the *terminal* device. Typically, the host device is a computer; the terminal device is a remote terminal or peripheral.

Transmitter protocol is used for batch, or block, information transmission. Both the host and the terminal must be set to use the same block size—the number of bytes in a transmission.

If the host is ready to send data to the terminal, the host sends an ENQ to check if the terminal is ready to receive a block of data. When the terminal is ready, it sends an ACK to the host, and the host transfers one block of data and another ENQ. This cycle is repeated as required by the host. ENQ is called the *request* character, and ACK is called the *answer* character.

The host is presumed to have infinite input capability, so that the terminal can send data to the host without using any software handshake and without observing the block size—that is, the terminal can send any amount of data to the host any time it wants. Because this situation is not always practical, a host may control this type of transfer using a *prompt* character. Whenever the host is ready to accept data, it sends a prompt character to the terminal. The terminal then sends one *line* of data to the host. (A line of data is terminated by an end-of-line indicator—its length is not related to the block size.) The host sends another prompt when it's ready for another line.

The HP-IL/RS-232 interface can be either a host or a terminal when it uses transmitter protocol. It uses the request and answer characters specified in character registers C06 and C07.

If the interface is a host, it never uses a prompt character. That is, the interface doesn't automatically prompt the external device for data. However, you can program the HP-IL controller to send an appropriate prompt character to the external device via the interface. For example, the controller can check the interface's status for receive buffer empty, then send the prompt.

If the interface is a terminal and is enabled to detect a prompt character (R11-1,0 equal to "01"), the prompt character is defined by character register C09. The interface stops sending data to the host whenever it sends an end-of-line indicator, which is specified in character registers C02 and C03 (regardless of whether end-of-line delete and insert are enabled by R10).

If the interface is a terminal and is set to *not* use a prompt character (R11-1,0 equal to "00"), then it doesn't wait for a prompt before sending data and doesn't stop sending data at end-of-line indicators.

Transmitter protocol can be used with full-duplex and half-duplex devices.

Receiver Protocol. Receiver protocol is used by the receiving device to tell the transmitting device when to start and stop sending. It's called receiver protocol because the receiver controls the handshake.

Receiver protocol does not require the use of batch, or block, information transmission. However, block transmission can be used if desired.

With receiver protocol, the receiving device monitors its buffer. When its buffer is close to being full (or close to not being able to accept another block), the receiving device sends an XOFF to tell the sending device to stop sending. The sending device then suspends data transmission until the receiving device sends an XON. The sending device then resumes sending, starting with the character that was next in line when the XOFF was received. XON (or DC1) is called the *ready* character, and XOFF (or DC3) is called the *not ready* character.

When the HP-IL/RS-232 interface uses receiver protocol (R11-3 equal to "1"), it uses the ready and not ready characters specified in character registers C04 and C05, respectively.

Receiver protocol requires full-duplex devices and operation. Both devices can be using receiver protocol to control data that they're receiving.

Combined Protocol. Some devices use both transmitter protocol and receiver protocol. This enables them to initiate the transfer of data according to transmitter protocol and to stop and start the transfer as required by the receiver.

When the HP-IL/RS-232 interface uses transmitter and receiver protocols (R11-3,2 equal to "11"), it operates according to both of those protocols as described above, including the host/terminal and prompt options.

Buffer Control

The interface monitors the number of data bytes in its buffer to determine when to send the appropriate software handshake character. By not allowing the external device to send data until there is enough room in the receive buffer for the next block of data, the interface can prevent the loss of data.

For transmitter protocol, the host normally sends data to the terminal in blocks followed by an ENQ. The interface's receive buffer can hold up to 109 bytes. If the interface is a terminal and if the external device (host) sends data in maximum blocks of 96 characters, for example, then the interface will not send an ACK until there are at least 96 empty bytes in the receive buffer.

Likewise, for transmitter protocol, if the interface is a host, then the external device will be expecting to receive data in blocks that it can handle. If the external device is expecting a block of data of 96 characters, for example, then the interface will send its data in blocks of 96 characters, each followed by an ENQ. The interface will not send out another block until it receives an ACK from the external device. (An ENQ is not sent at startup or reset.)

The block size used by the interface with transmitter protocol is normally a value from 1 to 256 while the interface is a host and from 1 to 109 while the interface is a terminal. Refer to page 84 for information about specifying the interface's transmitter block size, which is stored in character register C08. Refer to the owner's manual of the external device to determine the block size required.

For receiver protocol, the device that is sending data may continue sending data until it receives an XOFF. However, many devices do not immediately stop sending data when they receive the XOFF. The device may send a block of data before it can actually stop sending. Thus it is important to reserve a block of empty buffer bytes that is large enough to accept the maximum block that the external device may continue sending upon receiving an XOFF.

The interface reserves a block of bytes in its receive buffer. When the number of empty bytes decreases to this block size, the interface sends an XOFF. An XON is sent after an XOFF when the receive buffer becomes empty. (An XON is not sent at startup or reset.)

The interface will send a maximum of three characters after it receives an XOFF.

The block size used by the interface with receiver protocol is normally a value from 1 to 108. If the external device may send 12 characters after receiving an XOFF, for example, then the block size can be any value of 12 or more without risking loss of data. Refer to page 84 for information about specifying the receiver block size, which is stored in character register C10.

RS-232 Transmission Frames

In RS-232 serial transmission, the data is sent in frames, or packets, and each bit within the frame is determined by a voltage level and a bit rate. The voltage levels used by your HP-IL/RS-232 interface are −8V nominal for a logical "1" and +8V nominal for a logical "0". In the inactive or quiet state, the line is held at logical "1". The bit rate is selectable, and both devices *must* be set to the same bit rate. This bit rate is referred to as bit transmission rate. Some manufacturers refer to this as baud rate, even though baud rate is formally the rate of analog signal transmission. (Refer to section 5, page 49, for more information on baud rates.) Because the baud rate is equal to the mean bit transmission rate for standard RS-232 communication, this manual uses the term baud rate to mean bit transmission rate.

It is important for serial transmissions that the receiving device knows when data is being transferred and when data being transferred is about to stop. This information is conveyed by start and stop bits. The start bit synchronizes the receiving device so it reads the data properly. The stop bit(s) notifies the device that the data frame has ended.

For example, if a data character or word requires 8 bits, then the serial data frame is 10 bits long: 1 bit for the start bit, 8 bits for the character, and 1 bit for the stop bit.

Additionally, the frame may contain a bit called the parity bit. Parity is one method of checking the accuracy of the transmitted frame. If all the bits that have a value of "1" were counted, the result would be either an odd number or an even number. If both devices require that each frame have an even number of "1"s, then the receiver would know an error occurred if it received a frame that had an odd number. The parity bit is for setting the even or odd condition of a frame. If you select odd parity and the frame had an odd number of "1"s,

then the parity bit would be set to "0". If the frame had an even number, then the parity bit would be set to "1", and the frame would now have an odd number. Additionally, you can select to have the parity bit always set to "1" or to "0". Both the interface and the external device *must* have the same parity option selected.

Due to the requirements of different Data Terminal Equipment and Data Communication Equipment, it sometimes becomes neccessary to have two stop bits. Your HP-IL/RS-232 interface has the ability to implement parity and two stop bits. You must have at least one stop bit, but you aren't required to have a parity bit. When you select the number of data bits (the word size) the parity option, and the number of stop bits, the total number of bits in the frame is determined.

The following table shows some examples of how the total number of bits is determined.

Start Bit	Data Bits	Parity Bit	Stop Bits	Total Bits
1	7	none	1	9
1	7	1	2	11
1	8	1	1	11
1	8	1	2	12

You should be sure that the interface's RS-232 frame is defined the same as that of the external device. If the interface receives an RS-232 frame that differs from the expected format (that is, the stop bits don't occur at the expected time), the data byte isn't stored by the interface—that byte is lost.

The method for selecting the number of bits in the word, the number of stop bits, and the parity option is discussed on pages 78 and 79.

End-Of-Line Indicators

In many instances, the character sequence used by the HP-IL controller as an end-of-line indicator may not be the sequence used or recognized by the RS-232 device. Likewise, the HP-IL controller may not recognize the sequence used as an end-of-line indicator by the RS-232 device.

Your HP-IL/RS-232 interface is able to detect these indicators and replace them or delete them. For data going from HP-IL to RS-232, the interface can insert an end-of-line indicator of up to two characters. For additional information, refer to page 45.

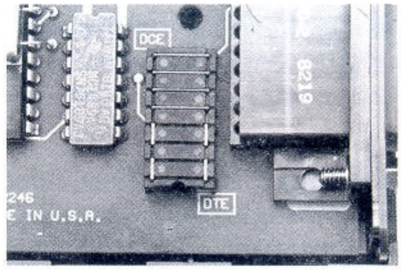
Changing the Configuration

The HP-IL/RS-232 interface—in its standard configuration—is a Data Terminal Equipment. Its internal RS-232 components control and interpret their signal lines as a normal DTE device. However, the *configuration selector* mounted internally on the printed-circuit board enables you to alter the standard connection between the internal components and the RS-232 receptacle on the back panel of the case. You can access the selector by removing the three case screws and opening the case.

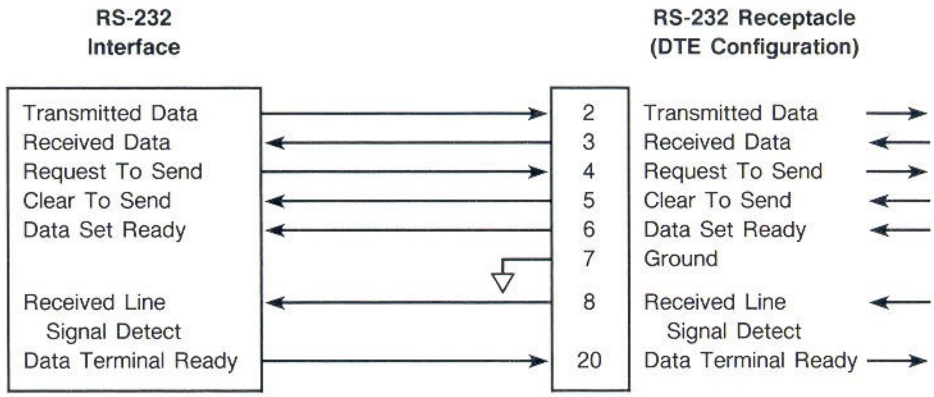
CAUTION

Use a small screwdriver to gently pry the jumper from its socket. If you attempt to remove the jumper by hand, you may damage its pins.

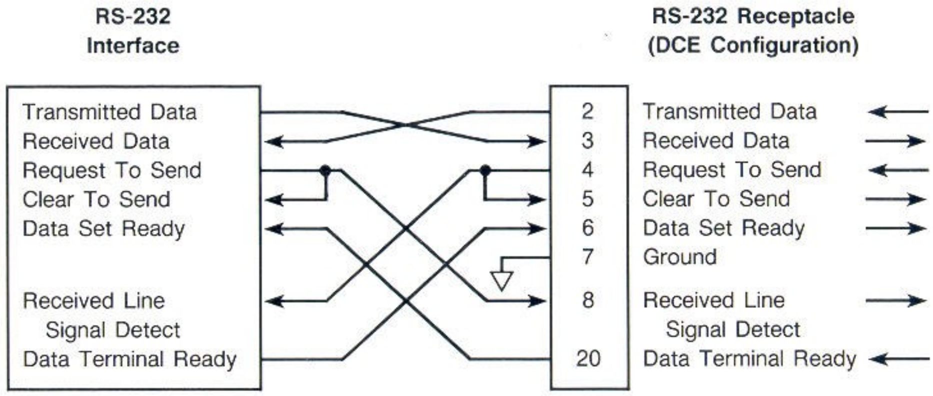
The selector consists of a socket and a number of jumpers that connect the internal signal lines to the RS-232 receptacle. By reversing the orientation of the jumper unit, the signal lines are reconnected in such a way that the interface emulates a Data Communication Equipment—that is, its RS-232 response resembles that of a DCE.



In the standard DTE configuration, the notched end of the jumper unit is located adjacent to the "DTE" marking on the printed-circuit board. The connections are shown below.



In the optional DCE configuration, the notched end of the jumper unit is located adjacent to the "DCE" marking on the printed-circuit board. The connections are shown below.



As an alternate, you can remove the jumper unit and connect the signals individually as required by your application. The diagram in appendix C defines the signal locations at the socket.

Section 3

HP-IL Operation

HP-IL is a friendly, low-cost method for enabling many devices to interact with each other. Information on HP-IL is serially processed and transmitted. All information travels in one direction around the interface loop. As each device on the loop receives an HP-IL message, the device determines if the message is for itself. If the message is for that device, the device takes the appropriate action and (in most cases) passes the message to the next device. If the message is not for that device, the device sends the message on to the next device. For additional information on the overall action of the interface loop, refer to the owner's manual for your controller or its HP-IL enhancement.

The interface provides complete compatibility with HP-IL. It interacts with other HP-IL devices by sending and receiving HP-IL messages on the interface loop. (Refer to the owner's manual for the HP-IL controller for information about controlling peripherals such as the HP-IL/RS-232 interface).

HP-IL Messages

There are four groups of HP-IL messages: *command* group, *ready* group, *identify* group, and *data/end* group. The command group is used by the controller to control the various peripherals on the loop. The ready group is for communication between the peripheral and the controller. The identify group is used for determining if a peripheral needs service. (Refer to "Parallel Polling," page 44, for additional information.) The data/end group is used for sending and receiving data and includes the data byte and end byte messages.

All HP-IL messages have the following 11-bit format:

$$C_2\,C_1\,C_0\quad D_7\,D_6\,D_5\,D_4\,D_3\,D_2\,D_1\,D_0$$

C_2 through C_0 are the control bits. They identify the message as one of the four types mentioned above. The meaning of the data bits D_7 through D_0 depends on the classification of the message as determined by the control bits.

The control bits are also used by the peripherals to tell the controller about the need for a service request. The definition of the control bits is listed in the following table.

Control Bits Decimal	Control Bits Binary	Definition	Meaning
0	000	Data Byte Message	The encoded message contains data.
1	001	Data Byte—Service Request	Same as Data Byte message, and a service request has been appended by a peripheral.
2	010	End Byte Message	The encoded message contains data and this is the last data byte in a group of data bytes.
3	011	End Byte—Service Request	Same as End Byte message, and a service request has been appended by a peripheral.
4	100	Command Message	The encoded message is from the command group.
5	101	Ready Message	The encoded message is from the ready group.
6	110	Identify Message	The encoded message is from the identify group.
7	111	Identify—Service Request	Same as Identify message, and a service request has been appended by a peripheral.

Whenever a Data Byte, End Byte, or Identify message passes through a peripheral, that peripheral can modify control bit C_0 to form a Data Byte—Service Request, End Byte—Service Request, or an Identify—Service Request message, respectively. Thus the controller can be notified that a peripheral needs service.

It is important to note that control bit C_0 can be modified to a service request condition by any peripheral on the loop. The peripheral to which the message was sent and the peripheral originating the message are not the only ones that can modify the control bits. Thus the controller knows only that one or more of the peripherals on the loop needs service—it does not know which one. To determine which peripheral needs service, the controller may then perform a parallel poll or a serial poll. (For further information on the operation of parallel polling and serial polling, refer to page 44.)

Responses to HP-IL Messages

The interface responds to HP-IL messages as described in the table below. The interface does not respond to any HP-IL messages that are not listed in the table. Except as noted in the table, each HP-IL message the interface receives is automatically sent to the next device in the loop. In general, the interface checks each message it initiates for transmission errors when the message comes back to the interface.

Responses to HP-IL Messages

HP-IL Message	Interface Response
Command Group	
Auto Address Unconfigure	Address set to 8.
Device Clear	Clears buffers and resets control and character registers to their default values (as at startup).
Device Dependent Listener 0-31	If listener, responds as described in table on page 28.
Device Dependent Talker 0-31	If talker, responds as described in table on page 28.
Enable Asynchronous Requests	Enables the interface to source an Identify—Service Request message on an idle loop. (Disabled by many command group messages—refer to page 44.)
Enable Listener Not Ready	Enables the interface to source a Not Ready For Data message if the interface is a listener and its transmit buffer is full. (Disabled by any other command group message.)
Go To Local	If listener, the interface responds to subsequent data bytes as data to be passed across the RS-232 lines. (Refer to "Using Remote Mode Instructions," page 36.)
Interface Clear	Talker or listener status removed and pending addressable message cleared (including device-dependent message).
Listen Address 0-31	If address matches,* device removed from talker status and device becomes a listener. If Remote mode is enabled, device changes to Remote mode. If message address is 31, device removed from listener status—same as Unlisten message.
Loop Power Down	No response. (Does not clear Enable Asychronous Request condition.)
No Operation	Clears Enable Asychronous Request condition.
Not Remote Enable	Removes the interface from Remote mode and sets it to Local mode. (Refer to "Local and Remote Modes" on page 28.)
Parallel Poll Disable	If listener, set to not modify subsequent Identify messages.
Parallel Poll Enable 0-15	If listener, set to modify subsequent Identify messages according to parallel poll conventions. (Refer to page 44.)
Parallel Poll Unconfigure	Set to not modify subsequent Identify messages.
Remote Enable	Enables the interface to begin operating in Remote mode whenever it next becomes a listener. (Refer to "Local and Remote Modes" on page 28.)
Secondary Address 0-30	Following a Talk Address or Listen Address message, if primary and secondary addresses match device's addresses, becomes a talker or listener.
Selected Device Clear	If listener, clears buffers and resets control and character registers to their default values (as at startup).
Talk Address 0-31	If address matches,* device removed from listener status and becomes a talker. If address doesn't match, device removed from talker status. If message address is 31, device is removed from talker status—same as Untalk message.
Unlisten	Device removed from listener status.
Untalk	Device removed from talker status.

Responses to HP-IL Messages (Continued)

HP-IL Message	Interface Response
Ready Group	
Auto Address 0-31	If device has earlier assigned address, no response. If message address is 31, no response. If message address less than 31 and device doesn't have earlier assigned address, device address is set to message address, increments message address by one, and passes revised message.
Auto Extended Primary 0-31	If device has earlier assigned address, no response. If message address is 31, no response. If not preceded by Auto Extended Secondary message, no response. If preceded by Auto Extended Secondary 31, no response. If preceded by Auto Extended Secondary less than 31, if message address less than 31, and if device doesn't have earlier assigned address, then device primary address is set to message address.
Auto Extended Secondary 0-31	If device has earlier assigned address, no response. If message address is 31, no response. If message address less than 31 and device doesn't have earlier assigned address, device secondary address set to message address, increments message address by one, and passes revised message. (Must be followed by Auto Extended Primary message to establish valid device address.)
End Of Transmission—Error	If talker, sent immediately for bad HP-IL error check.
End Of Transmission—OK	If talker, sent after last data byte or as described under "Interrupting Data Transfer," page 45.
Not Ready For Data	If talker, makes previous data byte the last byte sent. If listener and enabled to send this message, sent when a data byte fills the transmit buffer.†
Ready For Command	No response. (Not passed to next device until interface is ready for next command message.)
Send Accessory ID	If talker, sends one byte with the value 66.†
Send Data	If talker, begins sending contents of receive buffer, control registers, or character registers, as previously selected.†
Send Device ID	If talker, sends the ASCII-coded characters (bytes): HP82164A(CR)(LF).†
Send Status	If talker, sends four bytes of status. (Refer to page 41.)†
Identify Group	
Identify Identify—Service Request	If device set to respond by Parallel Poll Enable message, modifies message according to parallel poll setup and service request status. (Refer to page 44.) If service is required by interface, message modified to Identify—Service Request message.
Data/End Group	
Data Byte Data Byte—Service Request	If talker, sends next data byte.† If listener, accepts data byte and passes to next device. Data is normally sent to transmit buffer. If service is required by interface, message is modified to Data Byte—Service Request message.
End Byte End Byte—Service Request	If talker, sends next data byte. (Refer to "End-Of-Line Indicators" on page 45.)† If listener, accepts data byte and passes to next device. Data is normally sent to transmit buffer. (End-of-line sequence sent to RS-232 if enabled to do so.) If service is required by interface, message is modified to End Byte—Service Request message.

* For extended addressing, the message address must match the primary address. The response occurs only if the correct Secondary Address message follows.

† Indicates that a message different from the received message is sent to the next device in the loop.

Device-Dependent Messages

Device Dependent Listener messages and Device Dependent Talker messages (listed in the preceding table) are special HP-IL command messages whose meanings depend upon the device receiving them—the listener or the talker. When these messages are sent to the interface, they are referred to by names that correspond to the specific actions they cause. The device-dependent message numbers, names, and responses of your HP 82164A HP-IL/RS-232-C Interface are listed below.

Responses to Device-Dependent Messages

Message	Name	Interface Response
Device Dependent Listener:		
0	Set Control Registers	Up to 14 subsequent Data Bytes from HP-IL are stored in R00 through R13.*
1	Clear Transmit Buffer	Transmit buffer is cleared.
2	Set Character Registers	Up to 12 subsequent Data Bytes from HP-IL are stored in C00 through C11.*
3	Break On	Clears the transmit buffer and sends a continuous break signal (logical "0") to the external device on the Transmitted Data line. Continues until a Break Off instruction is received. (This condition may be recognized by some RS-232 devices.)
4	Break Off	Clears the transmit buffer and deactivates a break signal, enabling data to be sent to the external device.
5 - 31		No response.
Device Dependent Talker:		
0	Send Control Registers	Subsequent Send Data message causes the contents of R00 through R13 to be sent on HP-IL (14 Data Bytes).†
1	Clear Receive Buffer	Receive buffer is cleared.
2	Send Character Registers	Subsequent Send Data message causes the contents of C00 through C11 to be sent on HP-IL (12 Data Bytes).†
3 - 31		No response.

* The interface remains set to update the register contents until it has updated all registers or is next made a listener.

† The interface remains set to send the register contents until it sends an End Of Transmission message—as it does after it sends the contents or after it receives a Not Ready For Data message.

Local and Remote Modes

Your HP-IL/RS-232 interface has two operating states. The first state is called Local mode. In Local mode the interface's main function is to pass data from HP-IL devices to an RS-232 device. The second state is called Remote mode. In Remote mode the interface's main function is to set the interface's control and character registers using incoming data bytes as instructions.

In Local mode the interface responds to instructions from its ("local") keyboard and to HP-IL messages; data bytes are simply passed through to the RS-232 interface.

The Remote capability of the interface allows the interface's control and character registers to be set using data bytes, rather than requiring HP-IL device-dependent command messages. In Remote mode the interface interprets incoming data bytes as instructions from a remote source; the data bytes are not passed through to the RS-232 interface. Usually, the remote source is the HP-IL controller. For detailed information on how the interface responds to data bytes in Remote mode, refer to page 36.

There are a few precautions you need to be aware of to properly use the remote and local capabilities of your interface. These precautions are described next.

When your interface is turned on or the RESET key is pressed, the interface is set to operate in Local mode. To set the interface to Remote mode, the controller must send the HP-IL Remote Enable command message.

Remote Enable is a universal command, which means that it effects all devices on the loop. All devices on the loop that have the capability to operate in Remote mode are then ready to go into Remote mode. To set the interface into Remote mode, the controller must send the interface its Listen Address message—the Listen Address message that matches the interface's loop address. This makes the interface a listener, and it then begins operating in Remote mode.

Note that Remote Enable only sets the interface into a transitional state that *enables* it to go into Remote mode. Until the interface becomes a listener, the interface continues to operate in Local mode. This procedure is the only way to get the interface into its Remote mode.

There are two command messages that set the interface back into Local mode: Go To Local and Not Remote Enable. Go To Local is a device specific command, which means that only devices that are currently listeners will respond to the message. If your interface is a listener and it receives a Go To Local message, it changes back to Local mode. Remember that your interface is still remote enabled, so if it becomes a listener again, it will then change back to Remote mode.

The Not Remote Enable command not only puts the device back into Local mode, but it also disables the device from going back into Remote mode if it is made a listener. Not Remote Enable is a universal command that affects all devices on the loop that can respond to local and remote commands.

When the interface is in Remote mode it can be a talker, listener, or neither (idle). The interface can be moved in and out of these listen and talk states and still remain in Remote mode. But to get the interface to respond to a Go To Local message, the interface must be a listener at the time it receives that message. The interface responds to Not Remote Enable when it is a talker, listener, or neither.

Section 4

Control Logic

The interface's control logic governs the interface's interaction with both HP-IL and RS-232. The interface uses 14 control registers and 12 character registers to set specific operating conditions in its control logic. You can modify these registers and thus specify how the control logic governs the interface.

Control Registers

The control registers determine the actions of the interface as it translates the HP-IL formatted information into RS-232 formatted information. The 14 control registers are labeled R00 to R13. All control registers use only the lower four bits (bits 3 through 0) to represent the controlling information. When you set the value of a control register, the upper four bits (bits 7 through 4) of the data byte are ignored. When the interface sends the contents of a control register, it sends a data byte with a pattern of 0011XXXX. This allows the interface to receive the displayable ASCII characters listed in the table below.

Control Register Contents		Data Byte Sent		
Binary	Decimal	Binary	Decimal	ASCII Character
0000	0	0011 0000	48	0
0001	1	0011 0001	49	1
0010	2	0011 0010	50	2
0011	3	0011 0011	51	3
0100	4	0011 0100	52	4
0101	5	0011 0101	53	5
0110	6	0011 0110	54	6
0111	7	0011 0111	55	7
1000	8	0011 1000	56	8
1001	9	0011 1001	57	9
1010	10	0011 1010	58	:
1011	11	0011 1011	59	;
1100	12	0011 1100	60	<
1101	13	0011 1101	61	=
1110	14	0011 1110	62	>
1111	15	0011 1111	63	?

If you want to obtain the actual numerical value of the register contents, you can subtract the constant 48 from the value of the data byte sent.

Just a few of the control registers are discussed in this section. For a discussion of each control register, refer to appendix D.

Note: In this manual, individual bits in a control register are indicated by appending the bit numbers to the register name. For example, bits 2 and 1 of control register R02 are indicated by R02-2,1.

Control registers R00, R01, and R02 determine the conditions that will cause the interface to send a service request on HP-IL. (Refer to HP-IL message format on page 25.)

R00 — Service Request Conditions (Default 0000, Value=0)*

Bit 3	Bit 2	Bit 1	Bit 0
HP-IL Service Requests 0=Disable 1=Enable	Data Error 0=Disable 1=Enable	Unused	Receive Buffer Overflow 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

* If the internal service-request switch is opened, the default is 1101 (value = 13). (Refer to page 43.)

Bit 3 enables the interface to modify an appropriate HP-IL message to indicate a service request condition for any status conditions enabled by control registers R00, R01, and R02. If this bit is equal to "0", *no* service requests will be sent on HP-IL.

Bit 2 enables the interface to indicate a service request condition whenever any of these conditions occurs on RS-232: a parity error, a frame error, an overrun condition, or a receive buffer overflow condition. Requires that bit 3 be equal to "1" for the service request to be sent on HP-IL. (Refer to bits 7 through 4 of status byte 2, page 42.)

Bit 1 isn't used.

Bit 0 enables the interface to indicate a service request condition whenever the receive buffer is full and additional data has been received and lost. Requires that bit 3 be equal to "1" for the service request to be sent on HP-IL.

R01 — Service Request Conditions (Default 0000, Value=0)*

Bit 3	Bit 2	Bit 1	Bit 0
Receive Buffer Full 0=Disable 1=Enable	Receive Buffer Not Empty 0=Disable 1=Enable	Transmit Buffer Not Full 0=Disable 1=Enable	Transmit Buffer Empty 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

* If the internal service-request switch is opened, the default is 0101 (value = 5). (Refer to page 43.)

Bit 3 enables the interface to indicate a service request condition whenever the receive buffer is full. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 2 enables the interface to indicate a service request condition whenever the receive buffer is not empty. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 1 enables the interface to indicate a service request condition whenever the transmit buffer is not full. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 0 enables the interface to indicate a service request condition whenever the transmit buffer is empty. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

R02 — Service Request Conditions (Default 0000, Value=0)*

Bit 3	Bit 2	Bit 1	Bit 0
Unused	Manual Service Request 0=Disable 1=Enable	Auto-disconnect 0=Disable 1=Enable	Break Received 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

* If the internal service-request switch is opened, the default is 0010 (value = 2). (Refer to page 43.)

Bit 3 isn't used.

Bit 2 enables the interface to indicate a service request condition whenever the MSRQ key is pressed. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 1 enables the interface to indicate a service request condition whenever it has discontinued all communication on RS-232 (autodisconnect). Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 0 enables the interface to indicate a service request condition whenever it receives a break signal from the external device. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Control register R06 determines the number of stop bits and data bits used by the interface and whether it indicates parity errors.

R06 — Word Length and Parity (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
Number of Stop Bits 0=1 Bit 1=2 Bits	Number of Data Bits 00=8 Bits 01=7 Bits 10=6 Bits 11=5 Bits		Show Parity Error 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

Bit 3 specifies the number of stop bits that are sourced by the interface and that are expected by the interface on received data.

Bits 2 and 1 specify the number of bits that comprise the data character part of the transmission frame. (Refer to page 20 for additional information.)

Bit 0 enables the interface to indicate HP-IL data bytes that have RS-232 parity errors. This can be shown only when no more than seven bits are actually meaningful in the RS-232 data word. Eight bits may be used in the RS-232 data word, but the eighth bit (bit 7) must be "0". (Bit 7 is always "0" if you are using the properly defined ASCII character set.) If bit 7 is normally "0", then when Show Parity Error is enabled, the interface will set HP-IL bit D_7 to "1" whenever a byte with an RS-232 parity error is sent on HP-IL. For example, suppose the character "A" (value 65) is received on RS-232 with a parity error. The interface will then send this byte on HP-IL as the Data Byte message 000-11000001 (value 193). This option requires that R08-1 be equal to "1".

Control register R07 controls the RS-232 bit transmission rate (baud rate) of the interface.

R07 — Bit Transmission Rate (Default 1110, Value=14)

Bit 3	Bit 2	Bit 1	Bit 0
Bit Transmission Rate* 0000= 0 bps† 0001= 50 bps 0010= 75 bps 0011=110 bps 0100=135 bps 0101=150 bps 0110=300 bps 0111=600 bps		1000= 1200 bps 1001= 1800 bps 1010= 2400 bps 1011= 3600 bps 1100= 4800 bps 1101= 7200 bps 1110= 9600 bps 1111=19200 bps	
Value=8	Value=4	Value=2	Value=1

* Some manufacturers may refer to this as baud rate. (Refer to page 49.)

† The interface will neither send nor receive RS-232 data.

Bits 3 through 0 determine the rate at which the interface will send out information and read incoming information.

Control register R08 determines whether the interface will use a parity bit and whether that bit will be set according to even, odd, always 1, or always 0 conventions. The ability of the interface to echo back to the external device the characters as they are received is also set by this register.

R08 — Parity and Echo (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
Parity Select 00=Odd 01=Even 10=Always 1 11=Always 0		Parity Bit 0=Not present 1=Present	Echo 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

Bits 3 and 2 specify which parity option the interface uses. For detailed information on parity, refer to page 20.

Bit 1 enables or disables the parity option selected by bits 3 and 2. If bit 1 is equal to "1", the interface inserts a parity bit into its transmission frame and interprets the bit before the stop bit(s) as a parity bit. If bit 1 equals "0", then no extra bit is inserted and all bits except for start and stop bits are interpreted as data bits.

Bit 0 enables the interface to immediately send back—echo—the characters as they are received. This provides an additional means for error checking. If the characters that are echoed back are the ones used for displaying the information on the external device, then the display determines whether the interface is properly receiving the data. This option requires a full-duplex device. In addition, the controller should not send any data to the interface from HP-IL because it would disrupt the echo operation when sent on RS-232.

Character Registers

The interface has the capability to detect and delete certain characters or character sequences. Additionally, the interface can insert certain characters or character sequences on HP-IL or RS-232. These character sequences normally relate to end-of-line indicators. Other registers specify special characters used for software handshakes. The character registers enable you to set the characters that will be used. For a more detailed discussion, refer to appendix D.

Setting the Control and Character Registers

All of the conditions specified in the control registers must be properly set before the interface and the external device will fully operate the way you would like. Some conditions specified in the control registers, if not set properly, can be detected and corrected *after* your interface and external device are functioning. However, the following conditions must be properly set in both the interface and the external device *before* your interface and external device can communicate:

- Number of bits per word—control register R06.
- Number of stop bits—control register R06.
- Parity option—control register R08.
- Hardware handshake options—control register R09.
- Bit transmission rate (baud rate)—control register R07.
- Software handshake options—control register R11.

There are two ways to set and read the control and character registers: using device-dependent messages and using Remote mode instructions. For additional information about Local and Remote modes, refer to page 28.

Using Device-Dependent Messages

The interface is designed to respond to device-dependent messages only while it is in Local mode. The Device Dependent Listener 0 message is used to write information to the control registers; the Device Dependent Listener 2 message is used to write information to the character registers. The Device Dependent Talker 0 and 2 messages are used to read the control and character registers, respectively.

Described below is a sequence of HP-IL messages that illustrate how the controller might write to or read from the control registers. Only the core commands are illustrated, and they are illustrated in their proper order. The total sequence of all HP-IL messages that are required for proper HP-IL protocol is *not* described. Additionally, the individual controller may automatically handle some or all of the steps given below. Consult the owner's manual for your controller and the appendixes of this manual to find out how you might perform these operations.

To write to the control registers, you might use the sequence give below. For the purpose of this example, the interface is device 3 on the loop.

HP-IL Message	Response
Unlisten	Removes all listeners on the loop.
Listen Address 3	Makes the interface a listener.
Device Dependent Listener 0	Prepares the interface to use the subsequent Data Byte messages to sequentially set the control registers.
Data Byte 12	Sets control register R00 to 12 (1100), which enables the interface to send service requests on HP-IL, including when a data error occurs.
Data Byte 6	Sets control register R01 to 6 (0110), which enables a service request to be sent on HP-IL when the transmit buffer is full and when the receive buffer has data in it.
⋮	
Data Byte 4	Sets control register R13 to 4 (0100), which enables the interface to initiate an autodisconnect when the RS-232 device sets Data Set Ready line false.
End Of Transmission–OK	Tells the interface that no more data is coming and no transmission errors were detected.
Unlisten	Removes the interface from its listener state.

To read the control registers using the Device Dependent Talker 0 message, you might use the sequence listed below. Again the interface is device 3 on the loop.

HP-IL Message	Response
Unlisten	Removes all devices from listener state, so the controller is the only device that will process the Data Byte messages that the interface will send.
Talk Address 3	Makes the interface the talker on the loop. Any other talker is removed from the talker state.
Device Dependent Talker 0	Prepares the interface to sequentially send the values of the control registers.
Send Data	Tells the interface that it should now begin sending the Data Byte messages.
Data Byte 60	The lower four bits (refer to page 31) indicate that control register R00 has a value of 12 (1100), which indicates that the interface is enabled to send service requests on HP-IL, including when a data error occurs.
Data Byte 54	The lower four bits indicate that control register R01 has a value of 6 (0110), which indicates that the interface is enabled to send a service request on HP-IL when the transmit buffer is full and when the receive buffer has data in it.
⋮	
Data Byte 52	The lower four bits indicate that control register R13 has a value of 4 (0100), which indicates that the interface is enabled to initiate an autodisconnect when the RS-232 device sets Data Set Ready line false.
End Of Transmission–OK	Tells the controller that no more data is coming and no transmission errors were detected. (Sent by the interface.)
Untalk	Removes the interface from its talker state. (Sent by the controller.)

Using Remote Mode Instructions

The control and character registers can be set and read using Data Byte messages while the interface is in Remote mode. To set the interface to Remote mode, the controller must first send a Remote Enable message

followed by a Listen Address message that matches the interface's loop address. Refer to "Local and Remote Modes" on page 28.

Note: The Remote mode instruction characters refer to the ASCII character set as shown in appendix E. The interface actually responds to the numerical codes associated with the ASCII characters.

The Remote mode instruction set consists of a series of ASCII characters followed by one or more terminator characters. The format of any instruction falls into one of two forms. The first form consists of one or two uppercase letters and an additional character that specifies an option. The second form consists of two uppercase letters followed by four characters of your choice. The terminator characters consist of either a semicolon (;) or a line feed (LF) character.

The two possible formats are illustrated below.

SB2;
PC(ENQ)(ACK)(XON)(XOFF);

If the proper format is not followed, then an instruction syntax error is generated and the instruction is ignored. For example, the following list shows *improper* formats.

SB2,	Improper terminator.
PC(ENQ)(ACK);	Does not have all four characters specified.

You can also send out a string of commands by separating them with proper terminators. For example, you could send the following string:

SB2;C2(LF)PC(ENQ)(ACK)(XON)(XOFF);SW0;(CR)(LF)

The interface would accept this as a valid instruction sequence.

The space and carriage return (CR) characters are ignored and can be inserted anywhere in the string without affecting the interface's ability to decode the string, with following important exception: the FC*wxyz*, LC*wxyz*, and PC*wxyz* instructions use the four characters immediately following the "C" as the instruction characters. A space or CR in such a sequence would be used as a valid character.

When an improper instruction is received, the interface will ignore all subsequent characters until a valid terminator is received.

The following table lists the Remote mode instructions. The control register that is affected is listed in parentheses after the definition. Refer to the appropriate control register in appendix D for additional information about the response of the interface. A few instructions perform the same operations as device-dependent messages—the messages are listed in parentheses.

Remote Mode Instructions

Sequence	Definition
	Autodisconnect:
AE0	Disable (R13-2,1,0)
AE1	Enable for Clear To Send false (R13-0)
AE2	Disable for Clear To Send false (R13-0)
AE3	Enable for Received Line Signal Detect false (R13-1)
AE4	Disable for Received Line Signal Detect false (R13-1)
AE5	Enable for Data Set Ready false (R13-2)
AE6	Disable for Data Set Ready false (R13-2)
AE7	Enable for any line false (R13-2,1,0)

Remote Mode Instructions (Continued)

Sequence	Definition
	Break:
B0	Break off (R09-3)
B1	Break on (R09-3)
	Software Protocol:
C0	No protocol (R11-3,2,1,0)
C1	Transmitter protocol—terminal (R11-2,1)
C2	Receiver protocol (R11-3)
C3	Transmitter protocol—host (R11-2,1)
C4	Prompt enable (R11-0)
	Delete Characters:
DE0	Disable (R03-2,1,0)
DE1	Enable for DEL (R03-0)
DE2	Disable for DEL (R03-0)
DE3	Enable for NUL (R03-1)
DE4	Disable for NUL (R03-1)
DE5	Enable for selectable character (R03-2)
DE6	Disable for selectable character (R03-2)
DE7	Enable for DEL, NUL, selectable character (R03-2,1,0)
	Echo:
EE0	Disable (R08-0)
EE1	Enable (R08-0)
	Special-Function Characters:
FC*wxyz*	*w* specifies the transmitter block size (C08) *x* is the prompt character (C09) *y* specifies the receiver block size (C10) *z* is the delete character (C11)
	EOL Delete and Insert Characters:
LC*wxyz*	*w* is the first delete character (C00) *x* is the second delete character (C01) *y* is the first insert character (C02) *z* is the second insert character (C03)
	End-Of-Line Options:
LE0	Disable (R10-3,2,1,0)
LE1	Enable insert on RS-232 (R10-2)
LE2	Enable detect and delete from RS-232 only (R10-3,1)
LE3	Enable insert on HP-IL (R10-3,1)
LE4	Enable auto Request To Send (R10-0)
	Signal Line Control:
LI0	Disable (R04-3,2)
LI1	Set Data Terminal Ready true (R04-3,1)
LI2	Set Data Terminal Ready false (R04-3,1)
LI3	Set Request To Send true (R04-2,0)
LI4	Set Request To Send false (R04-2,0)
LI5	Disable Data Terminal Ready control (R04-3)
LI6	Disable Request To Send control (R04-2)
	Send Nulls at EOL:
NE0	Disable (R12-3)
NE1	Send one null (R12-3,2,1,0)
NE2	Send two nulls (R12-3,2,1,0)
NE3	Send three nulls (R12-3,2,1,0)
NE4	Send four nulls (R12-3,2,1,0)
NE5	Send five nulls (R12-3,2,1,0)
NE6	Send six nulls (R12-3,2,1,0)

Remote Mode Instructions (Continued)

Sequence	Definition
NE7	Send seven nulls (R12-3,2,1,0)
NE8	Send eight nulls (R12-3,2,1,0)
	Parity:
P0	Even parity (R08-3,2,1)
P1	Odd parity (R08-3,2,1)
P2	Always 0 (R08-3,2,1)
P3	Always 1 (R08-3,2,1)
P4	No parity (R08-3,2,1)
	Software Protocol Control Characters:
PC*wxyz*	*w* is the ready character (C04) *x* is the not ready character (C05) *y* is the request character (C06) *z* is the answer character (C07)
	Reset Buffer:
R0	Transmit buffer cleared (Device Dependent Listener 1)
R1	Receive buffer cleared (Device Dependent Talker 1)
	Baud Rate:
SB1	50 bps (R07-3,2,1,0)
SB2	75 bps (R07-3,2,1,0)
SB3	110 bps (R07-3,2,1,0)
SB4	135 bps (R07-3,2,1,0)
SB5	150 bps (R07-3,2,1,0)
SB6	300 bps (R07-3,2,1,0)
SB7	600 bps (R07-3,2,1,0)
SB8	1200 bps (R07-3,2,1,0)
SB9	1800 bps (R07-3,2,1,0)
SBA	2400 bps (R07-3,2,1,0)
SBB	3600 bps (R07-3,2,1,0)
SBC	4800 bps (R07-3,2,1,0)
SBD	7200 bps (R07-3,2,1,0)
SBE	9600 bps (R07-3,2,1,0)
SBF	19200 bps (R07-3,2,1,0)
	Service Request:
SE0	Disable all (R00-3,2,1,0; R01-3,2,1,0; R02-3,2,1,0)
SE1	Enable for transmit buffer empty (R00-3; R01-0)
SE2	Enable for transmit buffer not full (R00-3; R01-1)
SE3	Enable for receive buffer not empty (R00-3; R01-2)
SE4	Enable for receive buffer full (R00-3; R01-3)
SE5	Enable for receive buffer overflow (R00-3,0)
SE6	Enable for data error (R00-3,2)
SE7	Enable for break received (R00-3; R02-0)
SE8	Enable for autodisconnect (R00-3; R02-1)
SE9	Enable for manual service request (R00-3; R02-2)
	Hardware Handshake Lines:
SL0	Observe all lines (R09-2,1,0)
SL1	Ignore Clear To Send (R09-0)
SL2	Observe Clear To Send (R09-0)
SL3	Ignore Received Line Signal Detect (R09-1)
SL4	Observe Received Line Signal Detect (R09-1)
SL5	Ignore Data Set Ready (R09-2)
SL6	Observe Data Set Ready (R09-2)
SL7	Ignore all lines (R09-2,1,0)

Remote Mode Instructions (Continued)

Sequence	Definition
	Show Parity Error:
SP0	Disable (R06-0)
SP1	Enable (R06-0)
	Send Registers:
SR0	Send control registers (Device Dependent Talker 0)*
SR2	Send character registers (Device Dependent Talker 2)*
	Stop Bits:
SS0	1 stop bit (R06-3)
SS1	2 stop bits (R06-3)
	Word Length:
SW0	8 bits (R06-2,1)
SW1	7 bits (R06-2,1)
SW2	6 bits (R06-2,1)
SW3	5 bits (R06-2,1)

* Causes the interface to sequentially send the contents of the specified registers. After sending the "SR" instruction, you should make the interface a talker and send a Send Data message.

The following sequence of messages and instructions illustrate how you might use Remote mode intructions to control the interface. Note that when you are sending instructions to the interface, the interface must be a listener. Assume that the interface is device 3 on the loop.

Message or Instruction	Meaning
Unlisten	Removes all devices from listener state.
Remote Enable	Enables all devices on the loop to operate in Remote mode upon receiving the proper Listen Address message.
Listen Address 3	Makes the interface a listener and puts the interface in Remote mode.
SB3;SW0;P1;SS0(CR)(LF)	Sets the interface to use 300 baud, 8-bit data word, odd parity, and 1 stop bit.
C0;C2;SL0(CR)(LF)	Sets the interface to use only receiver protocol and observe all hardware handshake lines.
Unlisten	Removes the interface from listener state.
Not Remote Enable	Returns the interface to Local mode.

You can read the control and character registers while in Remote mode by using the SR0 and SR2 instructions. The following sequence will retrieve the contents of the control registers. Note that to send the SR0 instruction to the interface, the interface must be a listener—and that for the interface to send the contents of the control registers, the interface must be a talker.

Message or Instruction	Meaning
Unlisten	Removes all devices from listener state.
Remote Enable	Enables devices to operate in Remote mode upon receiving the proper Listen Address message.
Listen Address 3	Makes the interface a listener and puts it in Remote mode.
SR0(CR)(LF)	Tells the interface to prepare to send the contents of the control registers.
Talk Address 3	Removes the interface from listener state and makes it a talker.
Send Data	Instructs the interface to begin sending Data Byte messages.

Data Byte 60	The lower four bits (refer to page 31) indicate that control register R00 has a value of 12 (1100), which indicates that the interface is enabled to send service requests on HP-IL, including when a data error occurs.
Data Byte 54	The lower four bits indicate that control register R01 has a value of 6 (0110), which indicates that the interface is enabled to send a service request on HP-IL when the transmit buffer is full and when the receive buffer has data in it.
• • •	
Data Byte 52	The lower four bits indicate that control register R13 has a value of 4 (0100), which indicates that the interface is enabled to initiate an autodisconnect when the RS-232 device sets Data Set Ready line false.
End Of Transmission—OK	Tells the controller that no more data is coming and no transmission errors were detected. (Sent by the interface.)
Untalk	Removes the interface from the talker state. (Sent by the controller.)
Not Remote Enable	Returns the interface to Local mode.

Status

The interface maintains a four-byte record of its current condition in the status registers. The definitions of the status bytes are shown in the tables below. Normally, the status conditions in the status registers are updated whenever the interface's status changes. Also, bit 6 in status byte 1 (which indicates whether the interface has originated a service request on HP-IL) is cleared whenever a condition causing a service request condition is cleared or when the interface has sent its system status in response to a Send Status message.

Status byte 1 (the system status byte) can show only one condition at a time. The condition indicated is the highest priority condition that exists at the moment. Thus two or more system conditions may occur at the same time, but only one will be indicated. When a higher priority condition is cleared, then the next lower priority condition will be indicated.

Status Byte 1 Definition

Priority	Status Byte Decimal	Status Byte Binary*	Condition	Definition
1	132 or 196	1X000101	No Room	Autodisconnect occurred. (Refer to page 45.)
2	137 or 201	1X001001	Keyboard Input	MSRQ key was pressed since status last sent.
3	138 or 202	1X001010	Device Condition	Break signal was received on RS-232 since status last sent.
4	131 or 195	1X000011	Data Error	Received RS-232 data has had a parity error or bit pattern error, or has overflowed the receive buffer (HP-IL ← RS-232).
5	162 or 226	1X100010	Ready To Send Data (on HP-IL)	Receive buffer has data in it (HP-IL ← RS-232).
6	161 or 225	1X100001	Ready To Receive Data (on HP-IL)	Transmit buffer is not full (HP-IL → RS-232).
7	163 or 227	1X100011	Not Ready To Receive or Send Data (on HP-IL)	Transmit buffer is full (HP-IL → RS-232) and receive buffer is empty (HP-IL ← RS-232).

* The eight bits are shown in order—bit 7 (most significant) through bit 0 (least significant). An X indicates that bit 6 may be either a "0" or a "1". If bit 6 is a "1" (corresponding to the higher decimal value), the interface has originated a service request on HP-IL. Bit 6 is reset to a "0" when the status condition that caused the service request is cleared or when the system status byte has been sent following a Send Status message.

Status bytes 2, 3, and 4 are device status bytes. All status conditions in these bytes are shown simultaneously.

Status Byte 2 Definition

Bit Number	Bit Value*	Condition	Definition
7	128	RS-232 Parity Error	The interface has detected a parity error on RS-232.†
6	64	RS-232 Frame Error	An RS-232 frame has been received with a bit pattern error. (Such frames are deleted.)†
5	32	RS-232 Overrun	RS-232 data has been sent to the interface too fast and has been lost (HP-IL ← RS-232). (Causes an autodisconnect.)
4	16	Receive Buffer Overflow	The receive buffer is full and data has been lost (HP-IL ← RS-232).†
3	8	Receive Buffer Full	The receive buffer is full (HP-IL ← RS-232).
2	4	Receive Buffer Not Empty	Data is available in the receive buffer (HP-IL ← RS-232).
1	2	Transmit Buffer Not Full	The transmit buffer is not full (HP-IL → RS-232).
0	1	Transmit Buffer Empty	The transmit buffer is empty (HP-IL → RS-232).

* Add the bit values for all bits that are set (equal to "1") to find the decimal value of the status byte.

† Bit is reset to "0" after the interface has responded to a Send Status message.

Status Byte 3 Definition

Bit Number	Bit Value*	Condition	Definition
7	128	No Clear To Send Response	Request To Send is false and Clear To Send is true.
6	64	Manual Service Request	MSRQ key was pressed since status last sent.†
5	32	Autodisconnect	The interface has discontinued its RS-232 communication.
4	16	Break Received	The interface has received a break signal since status last sent.†
3	8	Remote Mode	The interface is operating in Remote mode.
2	4	Remote Mode Syntax Error	An error has been detected in the incoming sequence of Remote mode instructions.†
1	2	No Software Handshake	The software handshake is preventing data transmission.‡
0	1	No Hardware Handshake	The hardware handshake is preventing data transmission.

* Add the bit values for all bits that are set (equal to "1") to find the decimal value of the status byte.

† Bit is reset to "0" after the interface has responded to a Send Status message.

‡ Bit is reset to "0" when software handshake is redefined.

Status Byte 4 Definition

Bit Number	Bit Value	Condition	Definition
7	—		Reserved for future use.*
6	—		Reserved for future use.*
5	—		Reserved for future use.*
4	—		Reserved for future use.*
3	—		Reserved for future use.*
2	—		Reserved for future use.*
1	—		Reserved for future use.*
0	—		Reserved for future use.*

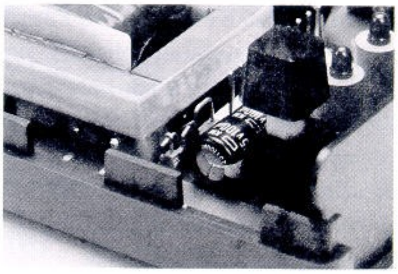
* Bit is set equal to "0".

Service Requests

A service request is an indicator used by a peripheral to notify the controller that it has some information for the controller, that it needs some information from the controller, or that it requires attention from the controller for some other reason.

Two types of conditions can cause the interface to initiate a service request (indicated by a control bit in an HP-IL Data Byte, End Byte, or Identify message): a *manual* service request and a *status* service request. A manual service request is initiated by pressing the MSRQ key. A status service request is initiated by the occurrence of a particular condition as indicated by the status register. The conditions that cause a service request are specified by control registers R00, R01, and R02. Bit 3 of control register R00 must be set equal to "1" in order to initiate any service request.

The default settings of control registers R0, R1, and R2 determine the service request conditions that are enabled at startup or reset. However, because HP-IL controllers differ in regard to their service request capabilities, the interface provides two sets of default settings for registers R00, R01, and R02. The interface has an internal switch on its printed-circuit board. When shipped, the switch is closed. In this condition, no service request conditions are enabled—that is, no service requests are indicated on HP-IL. If this switch is opened, five of nine conditions are enabled. (Refer to appendix D for additional information.) Of course, you can redefine which conditions initiate HP-IL service requests by using Remote instructions or a Device Dependent Listener 0 message.



If the interface is enabled to send a service request on HP-IL (R00-3) and a condition is enabled in register R00, R01, or R02, the occurrence of that condition will initiate a service request on HP-IL. When the next Data Byte, End Byte, or Identify message comes around the loop, the interface will modify that message to a Data Byte—Service Request, End Byte—Service Request, or Identify—Service Request message, respectively.

For example, assume the interface has been enabled to send service requests by having R00-3 equal to "1" and that the status condition Receive Buffer Full has been enabled (R01-3 equal to "1"). Also assume that the external device is sending data to the interface while the controller is busy with other devices. Then the following sequence of events could occur. Shortly after the receive buffer fills, an HP-IL Data Byte message comes around the loop. The interface would modify the control bits to a Data Byte—Service Request message. When the controller receives the data byte, it could check the control bits for a service request condition and find that a peripheral has requested service. The controller could then find that the interface has requested service by conducting either a serial poll or a parallel poll. Once the controller knows that the interface has requested service, the controller could read the status bytes and find that the interface's receive buffer is full.

The interface will continue to modify appropriate HP-IL messages to service request messages until the condition changes or until the interface has received a Send Status message.

The interface may also be enabled to send an Identify—Service Request message on an idle loop. To enable the interface to send an Identify—Service Request message, the controller would send an Enable Asynchronous Request message before it allowed the loop to become idle. On an idle loop the interface would not have an HP-IL message to modify. So the interface would send an Identify—Service Request message whenever it needed service. The controller should always handle an Identify—Service Request message in the same way, regardless of how the message is originated.

All but two "universal" command group messages disable the Enable Asynchronous Requests condition. Messages that disable this condition include Auto Address Unconfigure, Device Clear, Interface Clear, Local Lock Out, No Operation, Not Remote Enable, Parallel Poll Unconfigure, and Remote Enable. (Enable Asynchronous Requests and Loop Power Down do not disable this condition.)

Serial Polling

Serial polling is performed by making each peripheral a talker and checking its status register to see if it requested service. Because there can be only one talker on the loop at one time, each device must be separately checked. The HP-IL Send Status message is used in serial polling.

Serial polling is simpler than parallel polling (discussed next) for checking to see which device initiated a service request. But serial polling is also slower than parallel polling.

Parallel Polling

The interface can be enabled to respond to a parallel poll. A parallel poll allows the HP-IL controller to determine which devices require attention. When it receives an HP-IL Parallel Poll Enable message, the interface is set to respond in a particular way to subsequent parallel polls. The parallel poll consists of an HP-IL Identify message sent by the HP-IL controller. If the interface has been parallel poll enabled, it modifies all Identify messages according to the table below. Basically, for the first eight enable messages listed below, a "no service request" condition makes the designated bit a "1"; otherwise, the bit is not affected. For the last eight enable messages, a "service request" condition makes the designated bit a "1"; otherwise, the bit is not affected. In all cases, a "service request" condition is indicated by placing a "1" in the Service Request control bit in the Identify message. No other bits are affected by the interface.

If the interface receives a Parallel Poll Unconfigure message, or if the interface is a listener and receives a Parallel Poll Disable message, the interface won't respond to subsequent parallel polls—that is, it doesn't modify Identify messages.

Parallel Poll Response to Identify Message

Enable message . . .	designates bit . . .	and sets that Identify bit if . . .
Parallel Poll Enable 0	D_0	service is not requested*
Parallel Poll Enable 1	D_1	
Parallel Poll Enable 2	D_2	
Parallel Poll Enable 3	D_3	
Parallel Poll Enable 4	D_4	
Parallel Poll Enable 5	D_5	
Parallel Poll Enable 6	D_6	
Parallel Poll Enable 7	D_7	
Parallel Poll Enable 8	D_0	service is requested*
Parallel Poll Enable 9	D_1	
Parallel Poll Enable 10	D_2	
Parallel Poll Enable 11	D_3	
Parallel Poll Enable 12	D_4	
Parallel Poll Enable 13	D_5	
Parallel Poll Enable 14	D_6	
Parallel Poll Enable 15	D_7	

* Otherwise, the designated Identify bit isn't changed. Also, control bit C_0 is set if service is requested.

End-Of-Line Indicators

In its default condition, the interface is not set to detect characters or messages that indicate the end of a line of data (end-of-line indicators). Sequences of Data Bytes (and End Bytes) received from HP-IL are normally sent to the transfer buffer and then to RS-232 without being altered. Similarly, sequences received from RS-232 are sent to HP-IL without being altered. Of course, the external device or an HP-IL device may respond to a certain character or sequence as an end-of-line indicator, even though the interface isn't set to recognize it.

The table below lists the options for indicating the end of a line of data. Using these options, the interface can detect and delete end-of-line indicators. The interface can also be set to insert end-of-line indicators. This feature enables you to operate an external device with HP-IL, even if the end-of-line indicators are different.

End-Of-Line Indicators

Indicator Detected/Deleted	Indicator Added	Selected by
Output (HP-IL → RS-232)		
None	None	R10-2 = 0
End Byte	C02 and C03 on RS-232	R10-2 = 1
Input (HP-IL ← RS-232)		
None	None	R10-3 = 0
Specified RS-232 sequence: C00 and C01	End Byte on HP-IL	R10-3 = 1 R10-1 = 0
Specified RS-232 sequence: C00 and C01	C02 and C03 with End Byte on HP-IL	R10-3 = 1 R10-1 = 1

Interrupting Data Transfer

The HP-IL/RS-232 interface interrupts the transfer of data whenever its receive buffer is empty (it has no data to send) or its transmit buffer is full (it can't accept data).

For HP-IL ← RS-232 operation (the interface is a talker), the interface interrupts the transfer of data to HP-IL whenever the receive buffer becomes empty. If the external device stops sending data to the buffer (or fails to keep up with the HP-IL data rate), the interface will send an End Of Transmission message. In this situation, the external device causes the interruption.

In addition, for HP-IL ← RS-232 operation, if the interface receives a Not Ready For Data message on HP-IL, it interrupts data transfer with an End Of Transmission message. This interruption is under the control of the HP-IL controller—the external device doesn't initiate the action.

For HP-IL → RS-232 operation (the interface is a listener), the interface interrupts the data transfer when its transmit buffer is full—that is, it can't hold the byte just received. The interface suspends HP-IL operation (by not passing the Data Byte message to the next device) until it has room to store the byte.

However, for HP-IL → RS-232 operation, the interface can interrupt data transmission without suspending HP-IL operation if it has previously received an Enable Listener Not Ready message. When the transmit buffer fills (perhaps because the external device is not accepting data), the interface will send a Not Ready For Data message, allowing the controller to stop sending data to the interface.

Autodisconnect

An *autodisconnect* is an event that "permanently" interrupts the operation of the interface (that is, until the interface is reset). An event that causes an autodisconnect is an event that indicates the end of valid RS-232 communication.

Two types of autodisconnect events are defined for the interface: an RS-232 handshake line condition and a receive buffer overrun condition. If one or more input handshake lines (Data Set Ready, Received Line Signal Detect, or Clear To Send) are enabled to initiate an autodisconnect, then an autodisconnect will occur if the external device sets any of those lines false. (Lines are enabled by using the "AE" Remote instruction or by setting control register R13.) An overrun condition occurs when the interface can't process incoming RS-232 data fast enough to keep up. (This condition is indicated by bit 5 of status byte 2—it can often be avoided by setting the interface and external device to a baud rate that isn't too high.)

When an autodisconnect occurs, the interface stops all communication with HP-IL and RS-232. It sets all of its handshake lines false (unless they're being controlled by the controller), clears the transmit and receive buffers, and neither sends nor receives data on RS-232 and HP-IL. This condition is cleared when the interface receives a Device Clear (or Selected Device Clear) message or is reset.

Section 5

Modems

Introduction

One of the most common uses of RS-232 is interfacing to modems. The word modem is a contraction of the two words *mo*dulate/*dem*odulate. The modem converts digital signals to analog wave signals by modulating them. The modem then sends out the analog signals on a communication channel (typically a telephone line). When a signal comes in, the modem demodulates the incoming analog wave and converts it back to a digital signal.

Because the RS-232 standard deals specifically with connecting Data Terminal Equipment to Data Communication Equipment, connecting the interface to a modem is typically fairly simple. Most modems should connect directly to the interface. The only adjustments that should be required are properly setting the interface's control registers to match the corresponding settings on the modem or responding device.

There are basically two types of modems. One type is the bit modulator type. This type of modem takes each bit as it comes in, modulates the signal, and sends it out. With this type of modem you do not need to set any control features on the modem, nor do you need to set any control registers on your interface specifically for the modem—with one exception. The exception is the input bit transmission rate. Modems of this type match the bit transmission rate of the DTE because the bits go out at the same rate they come in. The exception is that most of these modems have only a certain range of bit transmission rates that they can properly operate within. As long as the interface and the answering modem are in this range, then the modem will function just fine.

The second type of modem requires that you interface to it. With this type of modem you must match your interface's bit transmission rate to the digital input and output bit transmission rate that the modem is expecting. Additionally, all other handshake options must be matched to the requirements of the modem. These types of modems frequently handle autoanswer connections.

Baud Rate

Strictly speaking, the baud rate is the bit or information transfer rate on an analog signal. The digital bit transmission rate does not need to match the analog bit transmission rate. On wide-bandwidth carrier waves it is often possible (and frequently done) to encode more than one bit of information in a single analog information unit. Thus the analog rate of transmission and the digital rate can differ significantly.

The term "baud" has become so common that most manufacturers use the term baud rate even for digital bit transmission rate. So in most cases you can equate bit transmission rate, bits per second (bps), and baud rate.

Acoustic Modems

The acoustic modem requires you to either manually place the call or manually answer the call. To place a call, you simply dial the number of the answering modem, and when you hear a high-pitched sound you place the handset in the acoustic coupler. To receive a call, you simply pick up the handset when the phone rings and place it in the acoustic coupler.

The connection to an acoustic modem is usually simply a matter of plugging the modem and the interface together (no special wiring needed) and setting the bit transmission rate, bits per word, parity, and handshake options. Additionally, the answering modem and answering DTE must be set to these same parameters.

Autoanswer Modems

The autoanswer modem can be as simple to use and connect to as the acoustic (manual answer) modem. The additional complication of automatically answering the call is usually handled by the modem itself. When a call comes in, the modem automatically establishes the connection and begins the data transfer. Therefore, it is important for the interface and modem to be properly connected and have the communications channels properly established before a call comes in.

Once the call is established, the interface must notify the controller that information is being received. One way for the interface to notify the controller is through a service request. (For more information on service requests, refer to page 43.) You may set the interface to send a service request when it receives data from the RS-232 device by setting control registers R00-3 and R02-2 equal to "1". Thus when data from RS-232 is received, the interface will send a service request on HP-IL. When the controller detects the service request and determines that the interface has requested service and has data from the RS-232 device, the controller and interface can begin passing information through the modem.

Section 6

Serial Printer Operation

Introduction

One common use of RS-232 is interfacing to serial printers. Because serial printers have been around much longer than the RS-232 standard and because the standard does not specifically address printers, printer manufacturers implement the standard differently, and some serial printers do not conform exactly to the RS-232 specifications.

Potential Problems

If you reexamine the definitions of Data Terminal Equipment and Data Communication Equipment on page 14, you will notice that it is not clear which definition includes printers. For this reason, some printer manufacturers use a pin configuration for a DTE, and others use a pin configuration for a DCE. Additionally, printers with a DTE pin configuration often have female connectors, and DCE configured printers often have male connectors. Thus, interfacing to RS-232 serial printers can be confusing.

A closer look at the definitions of the control handshake lines show that they are to be used for *passing control* of the communications link, *not* handshaking in terms of controlling the *passing of information*. For example, it is possible (and frequently true) that a printer can accept data faster than it can print it. So eventually its buffer will overflow. You might think that the printer could set the Clear To Send line false when its buffer is full. But Clear To Send, by strict defintion, cannot be set false before the DTE sets Request To Send false. Clear To Send just passes control of the communications link. RS-232 does not define a clear handshake for controlling the *passing of information*. So printer manufacturers have frequently selected lines to use for this function. You need to understand how a printer uses the control lines. Additionally, some printers require that Received Line Signal Detect be true before they will print any data.

Many manufacturers are beginning to use software handshakes to control the passing of information.

If you don't use a handshake for the transfer of data to a printer, use a baud rate that sends data more slowly than the printer prints it. An RS-232 word has about 10 bits, so the baud rate should be less than about 10 times the print rate (in characters per second). For example, if the print rate is 30 characters per second, the baud rate should be no more than 300 bits per second.

Connecting to an HP 2601A Daisywheel Printer

The HP 2601A Daisywheel Printer has some of the difficulties mentioned above. The pin configuration is that of a Data Terminal Equipment, although the connector is a female (indicating a DCE).

The signal lines that the printer uses are listed below.

Pin	Signal Name	Direction
1	Protective Ground	
2*	Transmitted Data	← Printer
3*	Received Data	→ Printer
4*	Request To Send	← Printer
5*	Clear To Send	→ Printer
6*	Data Set Ready	→ Printer
7*	Signal Ground	
8*	Received Line Signal Detect	→ Printer
11	Printer Ready	← Printer
20*	Data Terminal Ready	← Printer

* These are the only connections needed when interfacing to the HP-IL/RS-232 interface.

The printer sets its Request To Send and Data Terminal Ready lines true whenever the power is on. The printer expects its Data Set Ready line to be set true by the interface before the printer will send or receive data. The printer's Clear To Send line must be set true before the printer will send information to the interface. Also, the printer expects its Received Line Signal Detect line to be set true when the interface is sending a good signal.

Initially set the control switches on the printer to 300 baud, full-duplex, no parity, and DC1/DC3 (XON/XOFF) enable.

Because the printer has a DTE configuration, install the interface's internal configuration selector in the DCE position. This simplifies the electrical connection by making the pin assignments compatible. Although the connectors on the two units are physically compatible, use an extension to make the actual connection.

The following connections should be made for the interface and printer to function properly.

Interface Signal Name*	Interface Pin		Printer Pin	Printer Signal Name
Received Data*	2	←	2	Transmitted Data
Transmitted Data*	3	→	3	Received Data
Received Line Signal Detect*	4	←	4	Request To Send
	5	→	5	Clear To Send
Data Terminal Ready*	6	→	6	Data Set Ready
Signal Ground	7	—	7	Signal Ground
Request To Send*	8	→	8	Received Line Signal Detect
Data Set Ready*	20	←	20	Data Terminal Ready

* The listed signal names are those of the interface's internal signals. (Refer to page 22 for a diagram of the DCE connections.)

To properly configure the interface, ensure that it operates at 300 bits per second and uses no parity, one stop bit, an eight-bit word, and receiver software handshake. (Only the baud rate of 300 bits per second differs from the default control register conditions.) Set the control register listed below:

Control Register	Decimal Value	Binary Value
R07	6	0110

In the configuration described above, the printer is in a constant state of expecting data. When the interface is made a listener and data is sent on HP-IL, the interface will send the information out on RS-232, and the printer should print the data.

You can change this basic configuration in two ways. You can enable even or odd parity on both the printer and the interface, and you can select a different baud rate.

The printer can use either odd or even parity. Set the printer's parity enable switch (to enable parity) and the even parity switch (for even or odd parity). Set the interface's control register R06 to 0010 (value = 2) to specify a seven-bit word, using the eighth bit for a parity bit. Then set control register R08 to 0110 (value = 6) for even parity or to 0010 (value = 2) for odd parity.

The bit transmission rate can be set to any of the rates that are available on both the interface and the printer—1200 bits per second, for example. The interface and the printer are set for receiver protocol (XON/XOFF) software handshake, which prevents the printer's buffer from overflowing. Set the printer's switches to the proper baud rate, and set the interface's control register R07 to the value for the same baud rate.

Connecting to an NEC 3510 Spinwriter

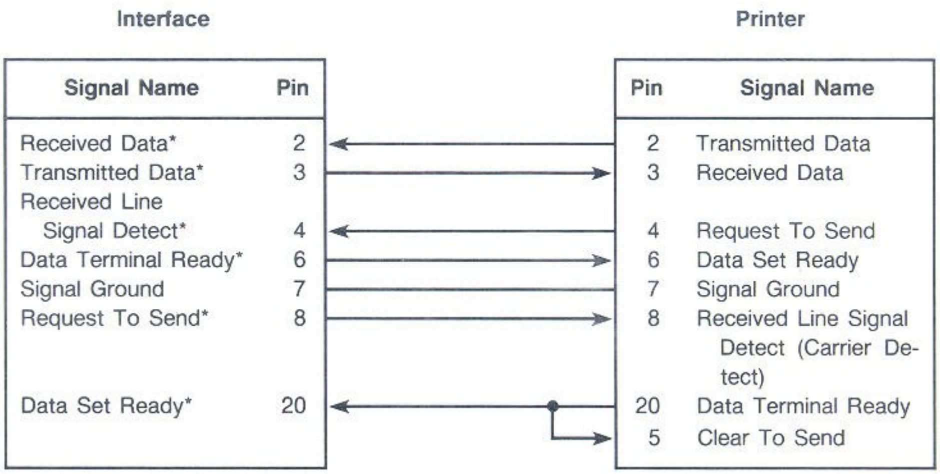
The NEC 3510 printer has a female connector and a Data Terminal Equipment pin configuration. The signal lines and pin assignments used by the printer are listed below.

Pin	Signal Name	Direction
2*	Transmitted Data	← Printer
3*	Received Data	→ Printer
4*	Request To Send	← Printer
5*	Clear To Send	→ Printer
6*	Data Set Ready	→ Printer
7*	Signal Ground	
8*	Received Line Signal Detect (Carrier Detect)	→ Printer
11	Reset	→ Printer
18	Keyboard Inhibit	→ Printer
19	Reverse Channel	← Printer
20*	Data Terminal Ready	← Printer
21	Print Inhibit	→ Printer
22	Buzzer	→ Printer
23	Paper Out/Ribbon End	← Printer
25	Interrupt/Break	← Printer

* These are the only connections needed when interfacing to the HP-IL/RS-232 interface.

The printer sets the Data Terminal Ready line true whenever the power is on and no error is detected. The printer sets Request To Send true after it sets Data Terminal Ready true and the interface sets the printer's Data Set Ready and Clear To Send lines true. Received Line Signal Detect (Carrier Detect) must be held true for the printer to work unless this line has been disabled by one of the printer's internal switches.

Because the printer has a DTE configuration, place the interface's internal selector in the DCE position. Then make the following connections. (You don't need to use the printer lines that aren't shown.)



* The listed signal names are those of the interface's internal signals. (Refer to page 22 for a diagram of the DCE connections.)

The simplest configuration for using this printer uses a baud rate of 300 bits per second, even parity, and receiver software handshake.

Set the printer's switches to 300 baud, even parity, and XON/XOFF enable. Set the interface's control registers as listed below—these registers set the interface to use one stop bit, seven-bit word, 300 baud rate, and even parity. (Other registers use their default values.)

Control Register	Decimal Value	Binary Value
R06	2	0010
R07	6	0110
R08	6	0110

Section 7

Connecting to Data Terminal Equipment

Introduction

Interfacing to a Data Terminal Equipment is generally not difficult. A device that is configured as a Data Terminal Equipment usually has a male connector with the proper DTE pin configuration. In this situation, install the interface's internal configuration selector in the DCE position. This allows the interface and DTE to be connected using the corresponding pins on each unit.

Interface

Signal Name*	Pin
Received Data*	2
Transmitted Data*	3
Received Line Signal Detect*	4
	5
Data Terminal Ready*	6
Signal Ground	7
Data Set Ready*	20

DTE Device

Pin	Signal Name
2	Transmitted Data
3	Received Data
4	Request To Send
5	Clear To Send
6	Data Set Ready
7	Signal Ground
20	Data Terminal Ready

* The listed signal names are those of the interface's internal signals. (Refer to page 22 for a diagram of the DCE connections.)

A data terminal often has an option for an input/output card that is wired and operates like a Data Communication Equipment. A data terminal with a DCE option can usually be connected to the interface in its standard DTE configuration (the internal selector in its DTE position).

For either connection, be sure to check for the proper baud rate, bits per word, parity, and handshake options.

Connecting to an HP Series 80 Personal Computer

You can connect your interface to an HP Series 80 Personal Computer using an HP 82939A Serial Interface with the standard option. The standard option provides a DCE configuration (female connector) that can connect to the HP-IL/RS-232 interface in its standard DTE configuration. Ensure that the HP-IL/RS-232 interface has its internal selector in the DTE position.

In its default condition, the serial interface operates with seven bits per word, odd parity, one stop bit, and 300 baud rate. To use these conditions, set the following control registers in the HP-IL/RS-232 interface to the values specified below.

Control Register	Decimal Value	Binary Value
R06	2	0010
R07	6	0110
R08	2	0010
R10*	4	0100

* This setting is optional. The HP-IL controller can send CR LF at the end of each string, instead.

The following simple program causes the Series 80 computer to use software receiver handshake, to test for data in its input register, and to display the data on the screen. The program also prompts for data to be transmitted to the interface. (While this program is running on the Series 80 computer, a similar program should be running on the HP-IL controller, so that the two devices can exchange data.)

`10 CONTROL 10,11;192`	Enables control of transmit flag.
`20 CONTROL 10,14;17,19`	Defines XOFF and XON characters.
`30 STATUS 10,10;A`	
`40 IF BIT(A,0)=0 THEN 70`	Tests for received data.
`50 ENTER 10;A$`	Reads data.
`60 DISP A$`	
`70 DISP 'MESSAGE TO SEND';`	
`80 INPUT A$`	
`90 OUTPUT 10;A$`	Sends data.
`100 GOTO 30`	

Connecting to an HP 3000 Computer

If you need to send information to an HP 3000 computer or need to receive information from an HP 3000, you can connect your HP-IL/RS-232 interface to the HP 3000 computer. Standard input control cards for the HP 3000 are structured as DCE, so the interface can be connected in its standard DTE configuration.

The default conditions for the HP-IL/RS-232 interface are compatible with most HP 3000 computer configurations. This means that you don't need to redefine any control or character registers for most installations. (If your system won't work with this configuration, check the RS-232 characteristics of your HP 3000 and compare them with the default conditions of the interface.)

Care, Warranty, and Service Information

Care of the Interface

The HP 82164A HP-IL/RS-232-C Interface contains sensitive electronic components that may be damaged by improper handling and use. Observe the following precautions to minimize the possibility of damage:

- When connecting wires or circuitry to the interface's RS-232 connector, be sure the external RS-232 connector is properly wired *before* it is plugged into the interface.
- Take precautions against damage to the interface's circuitry from electrostatic discharge.
- Observe the electrical specifications listed on page 71.
- Observe the temperature limits listed on page 73.

Verifying Proper Operation

If the interface's operation becomes disrupted for any reason, you can restore the interface to its startup condition by pressing the RESET key.

If at any time you suspect that your interface is not operating properly, you can verify its operation using the following test. This test checks the continuity of the interface loop and the operation of most of the interface's circuitry.

1. Perform the interface's self-test. (Connect only the ac adapter to the interface and press the RESET key.)
   - If the PWR and T/R lights turn on, and then the T/R light turns off after about 2 seconds, the tested portion of the interface is good.
   - If any other response occurs (particularly if the T/R light stays on), the interface requires service.
2. Connect only the interface and HP-IL controller in the interface loop.
3. Prepare the interface for testing:
   - Connect pin 2 to pin 3 at the RS-232 connector. This enables the interface to receive the data it sends on RS-232. (Don't connect an external device to the interface.)
   - Reset the interface to its default conditions by pressing the RESET key.
4. Using the controller, send one or more Data Bytes to the interface.
   - If the HP-IL messages (including Data Bytes) are passed around the loop and back to the controller, the interface and HP-IL cables have proper continuity.
   - If HP-IL messages do not return to the controller, the HP-IL continuity is bad. To determine the cause, try different cables or a different HP-IL peripheral. If HP-IL continuity is a problem for only the interface, then the interface requires service.
5. Using the controller, make the interface a talker and retrieve the previous Data Bytes from the interface.
   - If the retrieved Data Bytes match the original Data Bytes, the interface is good.
   - If the retrieved Data Bytes don't match the original Data Bytes, the interface requires service.

If this procedure indicates proper operation, but you still experience difficulty operating the interface with an external device, check the RS-232 configuration and connections (refer to appendix F) and verify the operation of the external device (refer to the manual for that device).

Limited One-Year Warranty

What We Will Do

The HP 82164A HP-IL/RS-232-C Interface is warranted by Hewlett-Packard against defects in materials and workmanship for one year from the date of original purchase. If you sell your unit or give it as a gift, the warranty is automatically transferred to the new owner and remains in effect for the original one-year period. During the warranty period, we will repair or, at our option, replace at no charge a unit that proves to be defective, provided you return the unit, shipping prepaid, to a Hewlett-Packard service center.

What Is Not Covered

This warranty does not apply if the product has been damaged by accident or misuse or as the result of service or modification by other that an authorized Hewlett-Packard service center.

No other express warranty is given. The repair or replacement of a product is your exclusive remedy. **ANY OTHER IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS IS LIMITED TO THE SPECIFIED DURATION OF THIS WRITTEN WARRANTY.** Some states, provinces, or countries do not allow limitations on how long an implied warranty lasts, so the above limitation may not apply to you. **IN NO EVENT SHALL HEWLETT-PACKARD COMPANY BE LIABLE FOR CONSEQUENTIAL DAMAGES.** Some states, provinces, or countries do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.

This warranty gives you specific legal rights, and you may also have other rights which vary from state to state, province to province, or country to country.

Warranty for Consumer Transactions in the United Kingdom

This warranty shall not apply to consumer transactions and shall not affect the statutory rights of a consumer. In relation to such transactions, the rights and obligations of Seller and Buyer shall be determined by statute.

Obligation to Make Changes

Products are sold on the basis of specifications applicable at the time of manufacture. Hewlett-Packard shall have no obligation to modify or update products once sold.

Warranty Information

If you have any questions concerning this warranty, please contact an authorized Hewlett-Packard dealer or a Hewlett-Packard sales and service office. Should you be unable to contact them, please contact:

- In the United States:

Hewlett-Packard
1000 N.E. Circle Blvd.
Corvallis, OR 97330
Telephone: (503) 758-1010
Toll-Free Number: (800) 547-3400 (except in
Oregon, Hawaii, and Alaska)

- In Europe:

Hewlett-Packard S.A.
150, route du Nant-d'Avril
P.O. Box CH-1217 Meyrin 2
Geneva
Switzerland
Telephone: (022) 83 81 11

Note: Do *not* send units to this address for repair.

- In other countries:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, California 94304
U.S.A.
Telephone: (415) 857-1501

Note: Do *not* send units to this address for repair.

Service

Hewlett-Packard maintains service centers in most major countries throughout the world. You may have your unit repaired at a Hewlett-Packard service center any time it needs service, whether the unit is under warranty or not. There is a charge for repairs after the one-year warranty period.

Hewlett-Packard products are normally repaired and reshipped within five (5) working days of receipt at any service center. This is an average time and could possibly vary depending upon the time of year and work load at the service center. The total time you are without your unit will depend largely on the shipping time.

Obtaining Repair Service in the United States

The Hewlett-Packard United States Service Center for the HP 82164A HP-IL/RS-232-C Interface is located in Corvallis, Oregon:

Hewlett-Packard Company
Corvallis Division Service Department

P.O. Box 999
Corvallis, Oregon 97339, U.S.A.

or

1030 N.E. Circle Blvd.
Corvallis, Oregon 97330, U.S.A.

Telephone: (503) 757-2000

Obtaining Repair Service in Europe

Service centers are maintained at the following locations. For countries not listed, contact the dealer where you purchased your unit.

AUSTRIA
HEWLETT-PACKARD Ges.m.b.H.
Kleinrechner-Service
Wagramerstrasse-Lieblgasse 1
A-1220 Wien (Vienna)
Telephone: (0222) 23 65 11

BELGIUM
HEWLETT-PACKARD BELGIUM SA/NV
Woluwedal 100
B-1200 Brussels
Telephone: (02) 762 32 00

DENMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 Birkerod (Copenhagen)
Telephone: (02) 81 66 40

EASTERN EUROPE
Refer to the address listed under Austria.

FINLAND
HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 Espoo 10 (Helsinki)
Telephone: (90) 455 02 11

FRANCE
HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F-91947 Les Ulis Cedex
Telephone: (6) 907 78 25

GERMANY
HEWLETT-PACKARD GmbH
Kleinrechner-Service
Vertriebszentrale
Berner Strasse 117
Postfach 560 140
D-6000 Frankfurt 56
Telephone: (611) 50041

ITALY
HEWLETT-PACKARD ITALIANA S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 Cernusco Sul Naviglio (Milan)
Telephone: (2) 90 36 91

NETHERLANDS
HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
N-1181 KK Amstelveen (Amsterdam)
P.O. Box 667
Telephone: (020) 472021

NORWAY
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 Oesteraas (Oslo)
Telephone: (2) 17 11 80

SPAIN
HEWLETT-PACKARD ESPANOLA S.A.
Calle Jerez 3
E-Madrid 16
Telephone: (1) 458 2600

SWEDEN
HEWLETT-PACKARD SVERIGE AB
Skalholtsgatan 9, Kista
Box 19
S-163 93 Spanga (Stockholm)
Telephone: (08) 750 20 00

SWITZERLAND
HEWLETT-PACKARD (SCHWEIZ) AG
Kleinrechner-Service
Allmend 2
CH-8967 Widen
Telephone: (057) 31 21 11

UNITED KINGDOM
HEWLETT-PACKARD Ltd
King Street Lane
GB-Winnersh, Wokingham
Berkshire RG11 5AR
Telephone: (0734) 784 774

International Service Information

Not all Hewlett-Packard service centers offer service for all models of HP products. However, if you bought your product from an authorized Hewlett-Packard dealer, you can be sure that service is available in the country where you bought it.

If you happen to be outside of the country where you bought your unit, you can contact the local Hewlett-Packard service center to see if service is available for it. If service is unavailable, please ship the unit to the address listed above under "Obtaining Repair Service in the United States." A list of service centers for other countries can be obtained by writing to that address.

All shipping, reimportation arrangements, and customs costs are your responsibility.

Service Repair Charge

There is a standard repair charge for out-of-warranty repairs. The repair charges include all labor and materials. In the United States, the full charge is subject to the customer's local sales tax. In European countries, the full charge is subject to Value Added Tax (VAT) and similar taxes wherever applicable. All such taxes will appear as separate items on invoiced amounts.

Products damaged by accident or misuse are not covered by the fixed repair charges. In these situations, repair charges will be individually determined based on time and material.

Service Warranty

Any out-of-warranty repairs are warranted against defects in materials and workmanship for a period of 90 days from date of service.

Shipping Instructions

Should your unit require service, return it with the following items:

- A completed Service Card, including a description of the problem and system setup when the problem occurred.
- A sales receipt or other documentary proof of purchase date if the one-year warranty has not expired.

The product, the Service Card, a brief description of the problem and system configuration, and (if required) the proof of purchase date should be packaged in the original shipping case or other adequate protective packaging to prevent in-transit damage. Such damage is not covered by the original warranty; Hewlett-Packard suggests that you insure the shipment to the service center. The packaged unit should be shipped to the nearest Hewlett-Packard designated collection point or service center. Contact your dealer directly for assistance. (If you are not in the country where you originally purchased the unit, refer to "International Service Information" above.)

Whether the unit is under warranty or not, it is your responsibility to pay shipping charges for delivery to the Hewlett-Packard service center.

After warranty repairs are completed, the service center returns the unit with postage prepaid. On out-of-warranty repairs in the United States and some other countries, the unit is returned C.O.D. (covering shipping costs and the service charge).

Further Information

Service contracts are not available. Circuitry and designs are proprietary to Hewlett-Packard, and service manuals are not available to customers.

Should other problems or questions arise regarding repairs, please call your nearest Hewlett-Packard service center.

Potential for Radio/Television Interference (for U.S.A. Only)

The HP 82164A HP-IL/RS-232 Interface generates and uses radio frequency energy and, if not installed and used properly (that is, in strict accordance with the instructions in this manual), may cause interference to radio and television reception. It has been tested and found to comply with the limits for a Class B computing device in accordance with the specifications in Subpart J of Part 15 of FCC rules, which are designed to provide reasonable protection against such interference in a residential installation. However, there is no guarantee that interference will not occur in a particular installation. If the interface does cause interference to radio or television reception, which can be determined by turning the interface off and on, you are encouraged to try to correct the interference by one or more of the following measures:

- Reorient the receiving antenna.
- Relocate the interface with respect to the receiver.
- Move the interface away from the receiver.
- Plug the interface's power supply into a different outlet so that the power supply and the receiver are on different branch circuits.

If necessary, you should consult your sales representative or an experienced radio/television technician for additional suggestions. You may find the following booklet, prepared by the Federal Communications Commission, helpful: *How to Identify and Resolve Radio-TV Interference Problems*. This booklet is available from the U.S. Government Printing Office, Washington, D.C. 20402, Stock Number 004-000-00345-4.

Dealer and Product Information

For dealer locations, product information, and prices, please call (800) 547-3400. In Oregon, Alaska, and Hawaii, call (503) 758-1010.

RS-232 Technical Description

RS-232 Compatibility

"RS-232-C compatible" does not mean that every piece of equipment bearing that label will work perfectly with every other piece of equipment so labelled. What it does mean is that the equipment does not violate any of the specifications or characteristics set down in the standard known as Electronic Industries Association (EIA) RS-232-C. But within the scope of the RS-232 standard there is enough latitude to permit minor incompatibilities from one device to another, and these minor incompatibilities can cause unpleasant surprises for the unwary.

The reason for these incompatibilities can be traced back to the history prior to the development of the RS-232 standard. Serial transmission can be traced back to the telegraph. But because the telegraph used transmission lines dedicated to the telegraph, there was no need to be compatible with signal types. But as the teletype was introduced, and as people wanted to communicate over longer distances, they naturally turned to the telephone as the transmission line.

Phone companies were extremely unhappy at the prospect of finding all kinds of strange signals in their networks. The effect of all these signals on the networks was unknown as the networks were designed to carry analog (voice) signals. Additionally, voltage interfacing requirements ranged anywhere from 6V to 140V, depending on the equipment manufacturer.

As a result, the Electronic Industries Association established the RS-232 standard. But because there already existed so many different types of equipment, the standard was not made explicit for all aspects of the RS-232 interface.

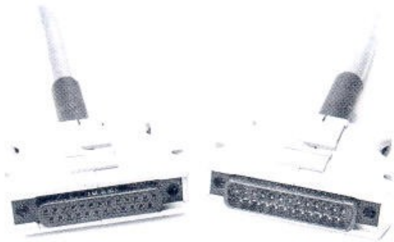
The standard governs the interfacing between Data Terminal Equipment and Data Communication Equipment using serial, binary interchange. The standard was originally formulated in 1963, and the latest revision has been in effect since 1969. The standard specifies:

- Mechanical characteristics of the interface.
- Electrical characteristics of the interface.
- A number of interchange circuits with descriptions of their functions.
- The relationship of interchange circuits to standard interface types.

The Comité Consultatif International Téléphonique et Télégraphique (CCITT) has established standards that correspond to RS-232-C. While these standards, CCITT V.24 and CCITT V.28, are very similar to RS-232-C, they are not identical. Because it does not make use of all the circuits defined in both RS-232-C and CCITT V.24, the HP 82164A HP-IL/RS-232-C Interface conforms to both RS-232-C and CCITT V.24 without any modification of the interface. The circuits that are utilized vary with different applications and with different equipment. The drivers and receivers used in the interface conform to voltage and other electrical specifications of both RS-232-C and CCITT V.28.

Mechanical Characteristics

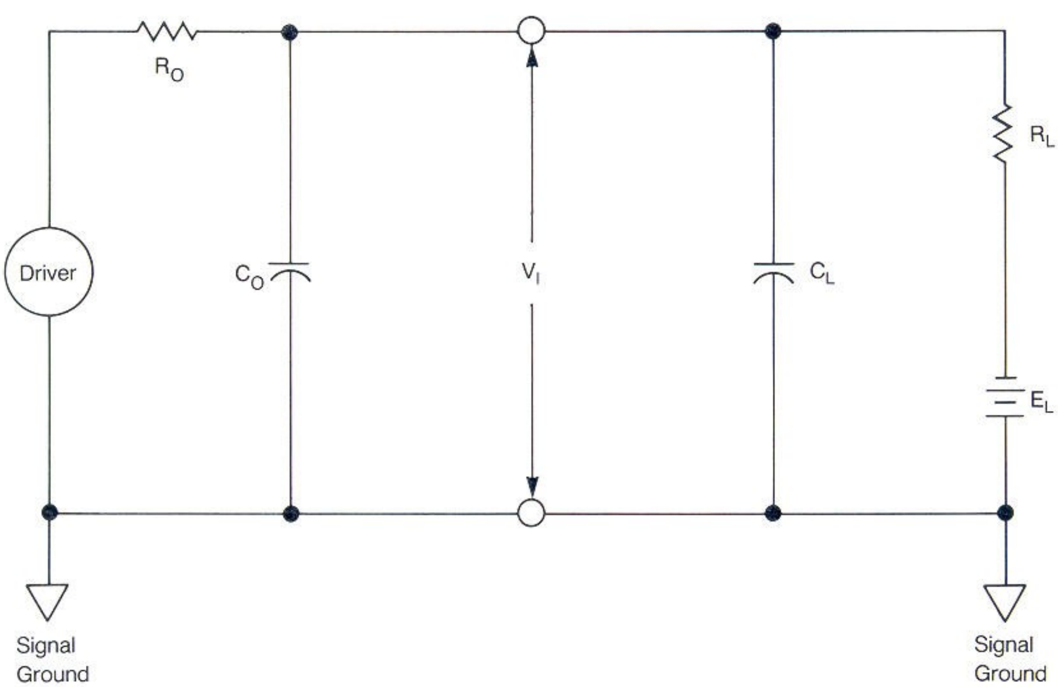
The standard gives definitions to 22 pins and designates 3 pins as unassigned, but does not specify a 25-pin connector. Although a particular connector is not defined, the industry has accepted the 25-pin D-subminiature connector shown below as a *de facto* standard. The standard does specify that the male connector is to be used with Data Terminal Equipment and that the female is to be used with Data Communications Equipment.



The length of the cable used by Data Terminal Equipment to connect to Data Communication Equipment should not be longer than 15 meters (50 feet). This length is based on a load capacitance at the interface point of 2500 picofarads, worst case. Longer cables are ofen used, especially in point-to-point configurations when it is known that the total load capacitance will not exceed the 2500 picofarads maximum.

Electrical Characteristics

A number of electrical parameters and limitations are defined by the RS-232 standard for each interchange circuit. They refer to the equivalent interchange circuit shown below. All voltage measurements are made at the interface point and with reference to the signal ground.



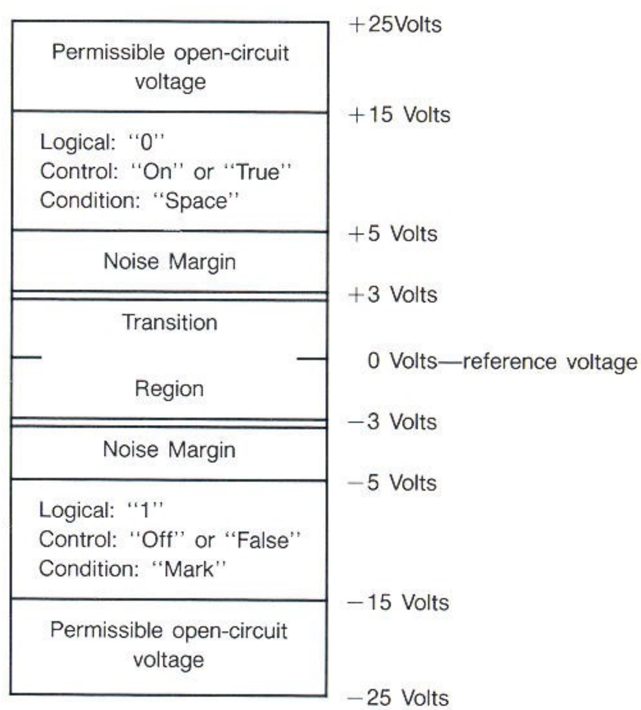
The electrical specifications are:

- Open circuit voltage (V_I) from the driver shall not exceed ±25 volts.
- The open circuit voltage (E_L) of the terminator shall not exceed ±2 volts.
- The total capacitance (C_L) of the terminator shall not exceed 2500 picofarads.
- The driver output voltage (V_I) must be between 5 and 15 volts when the total terminator input resistance (R_L) is between 3000 and 7000 ohms.

- The output impedance (R_O) of the driver circuit, when the driver power is off, shall be at least 300 ohms.
- The rate of change of the drive output voltage (V_I slew rate) shall not exceed 30 volts per microsecond.

In addition, several rules define the logic state indicated by voltage levels on the circuit. These rules are:

- A logical “1” (Mark or Off) is indicated when the voltage at the interface point is more negative than −3 volts.
- A logical “0” (Space or On) is indicated when the voltage at the interface point is more positive than +3 volts.
- To indicate a “1” (Mark or Off) signal condition, the driver shall assert a voltage between −5 volts and −15 volts.
- To indicate a “0” (Space or On) signal condition, the driver shall assert a voltage between +5 volts and +25 volts.



Note that these standards provide a 2-volt noise margin between the minimum driver voltage of 5 volts and the maximum undefined voltage of 3 volts. Other specifications that govern the transition region are:

- All interchange signals entering the transition region shall proceed to the opposite valid signal state. It shall not re-enter the transition region until the next significant change in signal state.
- While in the transition region, the direction of the voltage change must not reverse.
- The time required for a control signal to cross the transition region shall not exceed 1 millisecond.
- The time required for a data or timing signal to cross the transition region shall not exceed 1 millisecond or four percent of the nominal signal period, whichever is less.

RS-232 Function Table

This table contains the RS-232 functions, listed by pin number, as specified by the EIA RS-232-C standard. Note that these pins are defined from the viewpoint of the Data Terminal Equipment.

Pin	Function
1	**Protective Ground**. Electrical equipment frame and ac power ground.
2	***Transmitted Data**. Data transmitted to the DCE from the DTE.
3	***Received Data**. Data received by the DTE from the DCE.
4	***Request To Send**. Indicates to the DCE that the DTE is ready to transmit data.
5	***Clear To Send**. Indicates to the DTE that the DCE is ready to transmit data.
6	***Data Set Ready**. Indicates to the DTE that the DCE is not in a test mode and that it has power ON.
7	***Signal Ground**. Establishes common reference between the DCE and DTE.
8	***Received Line Signal Detect**. Indicates to the DTE that the DCE is receiving carrier signals from the sending DCE. (Also called Data Carrier Detect.)
9	Reserved for test.
10	Reserved for test.
11	Unassigned.
12	**Secondary Received Line Signal Detect**. Indicates to the DTE that the DCE is receiving secondary carrier signals from the sending DCE.
13	**Secondary Clear To Send**. Indicates to the DTE that the DCE is ready to transmit signals via the secondary channel.
14	**Secondary Transmitted Data**. Data transmitted to the DCE from the DTE on a secondary channel.
15	**Transmitter Signal Element Timing**. Signal from the DCE to the DTE to provide signal element timing information.
16	**Secondary Received Data**. Data from the DCE's secondary channel to the DTE.
17	**Receiver Signal Element Timing**. Signal to the receiving DTE to provide signal element timing information.
18	Unassigned.
19	**Secondary Request To Send**. Indicates to the DCE that the DTE is ready to transmit data via the secondary channel.
20	***Data Terminal Ready**. Indicates to the DCE that the DTE is ready to receive and transmit data.
21	**Signal Quality Detect**. Signal from the DCE telling whether a defined error rate in the received data has been exceeded.
22	**Ring Indicator**. Signal from the DCE indicating that a ringing signal is being received over the line.
23	**Data Signal Rate Selector**. Selects one of two signaling rates in DCE having two rates.
24	**Transmit Signal Element Timing**. Transmit clock provided by the DTE.
25	Unassigned.

* Function is implemented by the HP 82164A HP-IL/RS-232-C Interface.

Appendix C

Specifications

The tables that follow describe connector pin assignments of the HP 82164A HP-IL/RS-232-C Interface. The electrical and timing characteristics of the interface are fully compatible with the electrical and timing characteristics of the RS-232-C standard. (The electrical and timing requirements of the standard are described in appendix B.)

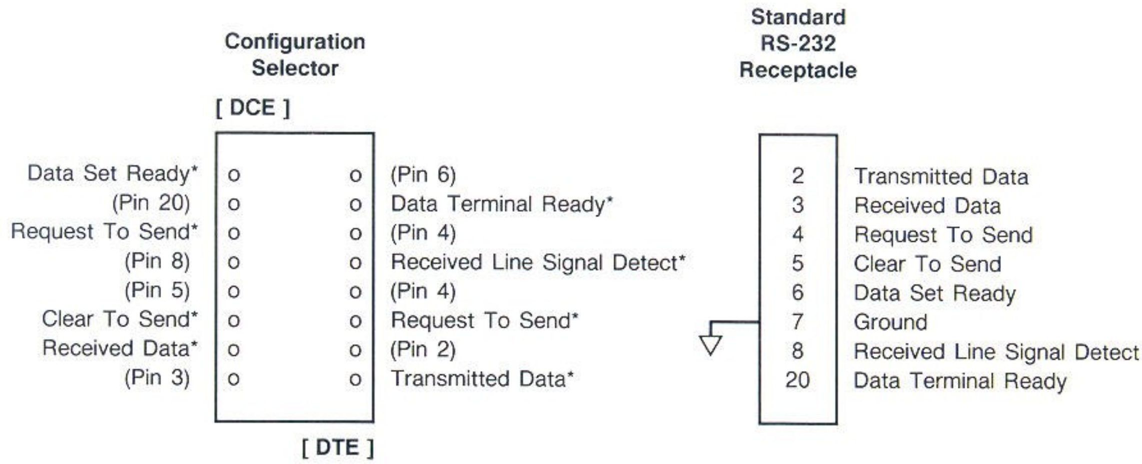
Temperature Limits

Operating	0° to 55° C (32° to 131° F)
Storage	−40° to 75° C (−40° to 167° F)

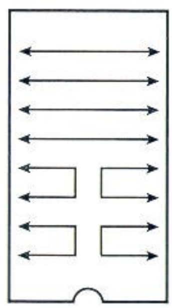
RS-232 Connector Pin Assignments

Pin	Signal
2	Transmitted Data
3	Received Data
4	Request To Send
5	Clear To Send
6	Data Set Ready
7	Signal Ground
8	Received Line Signal Detect (Carrier Detect)
20	Data Terminal Ready

The following diagram shows the connections between the interface's internal components and its RS-232 receptacle. The configuration selector (the socket on the printed-circuit board) is normally used with the jumper unit to set up the interface in a DTE configuration (its standard configuration) or in a configuration that emulates a DCE (with the jumper reversed). (Refer to "Changing the Configuration" on page 21.) As an alternative, you can remove the jumper unit and connect the signals individually as required by your application.



Configuration Jumper



Note: Install the jumper with its notch toward the desired option.

* Signal names refer to the *internal* signals used by the RS-232 circuitry. For the standard DTE configuration, the internal signals are connected to the RS-232 receptacle pins having the same signal names.

Register Descriptions

The HP 82164A HP-IL/RS-232-C Interface contains 14 control registers and 12 character registers. The control registers determine the way that the interface operates, as discussed throughout this manual. The character registers define the characters that will be used for inserting and deleting end-of-line characters. The tables below summarize the effects of the control registers.

When power is first supplied to the interface or when the RESET key is pressed, the control registers and character registers are initialized to the default values shown below. (The value for a register is determined by adding the indicated values of all bits that are "1".) The HP-IL controller can change the contents of the registers by using the HP-IL Device Dependent Listener 0 and the Device Dependent Listener 2 messages, or by using Remote mode instructions. (Refer to page 35 and page 36, respectively.)

Control Registers

The control registers allow you to control the operation of the interface by setting certain operating conditions within the interface. To set a control register you must set the desired bit pattern in that register. Refer to device-dependent messages and Remote mode instructions on pages 35 and 36 for details about setting these registers. For example, if you want to enable service requests for the data error condition, but disable service requests for the receive buffer overflow condition, then you can send the SE0 and SE6 Remote instructions, or you can use the Device Dependent Listener 0 message followed by a data byte with decimal value 12 (binary 00001100) to control register R00. (Note that the value you put into your controller to get your controller to send out the proper data byte depends upon the controller. A controller may require a decimal value, a special character, or a binary input—refer to your controller's owner's manual.)

Control Registers R00, R01, and R02

Control registers R00, R01, and R02 determine the conditions that will cause the interface to send a service request on HP-IL. (Refer to HP-IL message format on page 25.)

R00 — Service Request Conditions (Default 0000, Value=0)*

Bit 3	Bit 2	Bit 1	Bit 0
HP-IL Service Requests 0=Disable 1=Enable	Data Error 0=Disable 1=Enable	Unused	Receive Buffer Overflow 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

* If the internal service-request switch is opened, the default is 1101 (value = 13). (Refer to page 43.)

Bit 3. Bit 3 enables the interface to modify an appropriate HP-IL message to indicate a service request condition for any status conditions enabled by control registers R00, R01, and R02. If this bit is equal to "0", *no* service requests will be sent on HP-IL.

Bit 2. Bit 2 enables the interface to indicate a service request condition whenever any of these conditions occurs on RS-232: a parity error, a frame error, an overrun condition, or a receive buffer overflow condition. Requires that bit 3 be equal to "1" for the service request to be sent on HP-IL. (Refer to bits 7 through 4 of status byte 2, page 42.)

Bit 1. Bit 1 isn't used.

Bit 0. Bit 0 enables the interface to indicate a service request condition whenever the receive buffer is full and additional data has been received and lost. Requires that bit 3 be equal to "1" for the service request to be sent on HP-IL.

R01 — Service Request Conditions (Default 0000, Value=0)*

Bit 3	Bit 2	Bit 1	Bit 0
Receive Buffer Full 0=Disable 1=Enable	Receive Buffer Not Empty 0=Disable 1=Enable	Transmit Buffer Not Full 0=Disable 1=Enable	Transmit Buffer Empty 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

* If the internal service-request switch is opened, the default is 0101 (value = 5). (Refer to page 43.)

Bit 3. Bit 3 enables the interface to indicate a service request condition whenever the receive buffer is full. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 2. Bit 2 enables the interface to indicate a service request condition whenever the receive buffer is not empty. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 1. Bit 1 enables the interface to indicate a service request condition whenever the transmit buffer is not full. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 0. Bit 0 enables the interface to indicate a service request condition whenever the transmit buffer is empty. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

R02 — Service Request Conditions (Default 0000, Value=0)*

Bit 3	Bit 2	Bit 1	Bit 0
Unused	Manual Service Request 0=Disable 1=Enable	Auto-disconnect 0=Disable 1=Enable	Break Received 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

* If the internal service-request switch is opened, the default is 0010 (value = 2). (Refer to page 43.)

Bit 3. Bit 3 isn't used.

Bit 2. Bit 2 enables the interface to indicate a service request condition whenever the MSRQ key is pressed. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 1. Bit 1 enables the interface to indicate a service request condition whenever it has discontinued all communciation on RS-232 (autodisconnect). Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Bit 0. Bit 0 enables the interface to indicate a service request condition whenever it receives a break signal from the external device. Requires that bit R00-3 be equal to "1" for the service request to be sent on HP-IL.

Control Register R03

Control register R03 specifies the special characters that the interface will delete from data received on RS-232.

R03 — Delete Special Characters (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
Unused	Delete DEL from RS-232 0=Disable 1=Enable	Delete NUL from RS-232 0=Disable 1=Enable	Delete Selectable Character from RS-232 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 isn't used.

Bit 2. Bit 2 enables the interface to detect and delete the DEL (delete, character code 127) character from RS-232 data before passing the data to HP-IL. (DEL is sometimes referred to as "rubout.")

Bit 1. Bit 1 enables the interface to detect and delete the NUL (null, character code 0) character from RS-232 data before passing the data to HP-IL.

Bit 0. Bit 0 enables the interface to detect and delete a specified character from RS-232 data before passing the data to HP-IL. The character to be deleted is specified by character register C11.

Control Register R04

Control register R04 enables the Data Terminal Ready and Request To Send output lines to be individually controlled. Either one or both of these lines may be selected.

R04 — Signal Line Control (Default 1111, Value=15)

Bit 3	Bit 2	Bit 1	Bit 0
Data Terminal Ready Enable 0=Disable 1=Enable	Request To Send Enable 0=Disable 1=Enable	Data Terminal Ready 0=False 1=True	Request To Send 0=False 1=True
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 enables the controller to control the Data Terminal Ready line. If this bit is equal to "0", the Data Terminal Ready line goes false whenever the number of empty bytes in the receive buffer decreases to the number specified by C10, and then goes true when the number of empty bytes exceeds the number specified by C10.

Bit 2. Bit 2 enables the controller to control the Request To Send line. If this bit is equal to "0", the Request To Send line is false when the transmit buffer is empty and is true when the buffer isn't empty.

Bit 1. Bit 1 controls the state of the Data Terminal Ready line when bit 3 is equal to "1".

Bit 0. Bit 0 controls the state of the Request To Send line when bit 2 is equal to "1".

Control Register R05

Control register R05 indicates the status of the input signal lines.

R05 — Input Signal Line Status (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
Unused	Data Set Ready 0=False 1=True	Received Line Signal Detect 0=False 1=True	Clear To Send 0=False 1=True
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 isn't used.

Bits 2 through 0. Bits 2 through 0 indicate the status of particular input signals as shown in the table above. If the respective bit is equal to "1", then the external device is holding the line true. If the bit is "0", then the line is false. (Any value sent to this register is immediately updated to show the actual status of the lines.)

Control Register R06

Control register R06 determines the number of stop bits and data bits used by the interface and whether it indicates parity errors.

R06 — Word Length and Parity (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
Number of Stop Bits 0=1 Bit 1=2 Bits	Number of Data Bits 00=8 Bits 01=7 Bits 10=6 Bits 11=5 Bits		Show Parity Error 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 specifies the number of stop bits that are sourced by the interface and that are expected by the interface on received data.

Bits 2 and 1. Bits 2 and 1 specify the number of bits that comprise the data character part of the transmission frame. (Refer to page 20 for additional information.)

Bit 0. Bit 0 enables the interface to indicate HP-IL data bytes that have RS-232 parity errors. This can be shown only when no more than seven bits are actually meaningful in the RS-232 data word. Eight bits may be used in the RS-232 data word, but the eighth bit (bit 7) must be "0". (Bit 7 is always "0" if you are using the properly defined ASCII character set.) If bit 7 is normally "0", then when Show Parity Error is enabled, the interface will set HP-IL bit D_7 to "1" whenever a byte with an RS-232 parity error is sent on HP-IL. For example, suppose the character "A" (value 65) is received on RS-232 with a parity error. The interface will then send this byte on HP-IL as the Data Byte message 000-11000001 (value 193). This option requires that R08-1 be equal to "1".

Control Register R07

Control register R07 controls the RS-232 bit transmission rate (baud rate) of the interface.

R07 — Bit Transmission Rate (Default 1110, Value = 14)

Bit 3	Bit 2	Bit 1	Bit 0
Bit Transmission Rate*			
0000 = 0 bps†		1000 = 1200 bps	
0001 = 50 bps		1001 = 1800 bps	
0010 = 75 bps		1010 = 2400 bps	
0011 = 110 bps		1011 = 3600 bps	
0100 = 135 bps		1100 = 4800 bps	
0101 = 150 bps		1101 = 7200 bps	
0110 = 300 bps		1110 = 9600 bps	
0111 = 600 bps		1111 = 19200 bps	
Value = 8	Value = 4	Value = 2	Value = 1

* Some manufacturers may refer to this as baud rate. (Refer to page 49.)

† The interface will neither send nor receive RS-232 data.

Bits 3 through 0. Bits 3 through 0 determine the rate at which the interface will send out information and read incoming information.

Control Register R08

Control register R08 determines whether the interface will use a parity bit and whether that bit will be set according to even, odd, always 1, or always 0 conventions. The ability of the interface to echo back to the external device the characters as they are received is also set by this register.

R08 — Parity and Echo (Default 0000, Value = 0)

Bit 3	Bit 2	Bit 1	Bit 0
Parity Select 00 = Odd 01 = Even 10 = Always 1 11 = Always 0		Parity Bit 0 = Not present 1 = Present	Echo 0 = Disable 1 = Enable
Value = 8	Value = 4	Value = 2	Value = 1

Bits 3 and 2. Bits 3 and 2 specify which parity option the interface uses. For detailed information on parity, refer to page 20.

Bit 1. Bit 1 enables or disables the parity option selected by bits 3 and 2. If bit 1 is equal to "1", the interface inserts a parity bit into its transmission frame and interprets the bit before the stop bit(s) as a parity bit. If bit 1 equals "0", then no extra bit is inserted and all bits except for start and stop bits are interpreted as data bits.

Bit 0. Bit 0 enables the interface to immediately send back—echo—the characters as they are received. This provides an additional means for error checking. If the characters that are echoed back are the ones used for displaying the information on the external device, then the display determines whether the interface is properly receiving the data. This option requires a full-duplex device. In addition, when the interface is receiving data on RS-232, the controller should not send any data to the interface from HP-IL because it could temporarily disrupt the echo operation when sent on RS-232.

Control Register R09

Control register R09 controls whether a break signal is sent to the external device and selects signal lines that are used for the hardware handshake.

R09 — Break and Hardware Handshake (Default 0111, Value=7)

Bit 3	Bit 2	Bit 1	Bit 0
Transmit Break Signal 0=Off 1=On	Data Set Ready Ignore 0=Observe 1=Ignore	Received Line Signal Detect Ignore 0=Observe 1=Ignore	Clear To Send Ignore 0=Observe 1=Ignore
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 controls the interface's transmitted break signal. While this bit is equal to "1", the interface sends a continuous break signal to the external device, suspending data transmission to that device. While this bit is equal to "0", data can be sent to the external device according to the software and hardware handshakes.

Bits 2 through 0. Bits 2 through 0 select which signal lines will be used by the interface for its hardware handshake. For each bit that is equal to "1", the corresonding signal isn't checked by the interface and doesn't prevent the interface from assuming that the device is ready. For each bit that is equal to "0", the interface requires that the signal be true before it will perform the corresponding data transfer on RS-232. (Refer to page 17.)

Control Register R10

Control register R10 specifies the end-of-line options that the interface will use.

R10 — End-Of-Line Options (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
End-Of-Line Detect and Delete from RS-232 0=Disable 1=Enable	End-Of-Line Insert on RS-232 0=Disable 1=Enable	End-Of-Line Insert on HP-IL 0=Disable 1=Enable	Auto Request To Send 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 enables the interface to look for end-of-line characters on RS-232, to delete them, and to send the preceding byte as an End Byte message on HP-IL. (If an end-of-line indicator is not preceded by other data, the indicator is deleted and nothing is sent on HP-IL.) The end-of-line characters are specified in character registers C00 and C01.

Bit 2. Bit 2 enables the interface to insert end-of-line characters on RS-232 upon receiving an End Byte message on HP-IL. The characters are inserted after the data from the End Byte is sent. The characters are specified in character registers C02 and C03.

Bit 1. Bit 1 enables the interface to insert end-of-line characters on HP-IL upon detecting and deleting an RS-232 end-of-line indicator. The characters are specified in character registers C02 and C03. Bit 3 must be equal to "1" to enable this capability.

Bit 0. Bit 0 enables the interface to control the Request To Send line according to half-duplex conventions. If this bit is equal to "1", it causes the Request To Send line to go false whenever an end-of-line indicator (specified by registers C02 and C03) is sent on RS-232. (Refer to page 16 for additional information.) R04-2 and R09-0 must each be equal to "0" for this option to be available.

Control Register R11

Control register R11 allows you to select the software handshake protocols that the interface will use. (Refer to page 18 for detailed information.) The interface is enabled to start sending data on RS-232 without additional software handshake characters whenever this register is redefined.

R11 — Software Handshake (Default 1100, Value=12)

Bit 3	Bit 2	Bit 1	Bit 0
Receiver Protocol 0=Disable 1=Enable	Transmitter Protocol 0=Disable 1=Enable	Transmitter Protocol Option 0=Terminal 1=Host	Prompt 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 enables the interface to use the receiver protocol to control the passing of information across RS-232. (The receiver protocol characters are defined by character registers C04 and C05.)

Bit 2. Bit 2 enables the interface to use transmitter protocol to control the passing of information across RS-232. (The transmitter protocol characters are defined by character registers C06 and C07.)

Bit 1. Bit 1 selects which type of transmitter protocol the interface will use, if transmitter protocol is enabled. Bit 2 must be equal to "1" for this option to be valid.

Bit 0. Bit 0 enables the interface to use a prompt character when using transmitter protocol as a terminal. Bit 2 must be equal to "1" and bit 1 must be equal to "0" for this option to be valid. (The prompt character is defined by character register C09.)

Control Register R12

Control register R12 enables the interface to "pause" at each end-of-line indicator.

R12 — End-Of-Line Wait (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
Send Nulls After End-Of-Line Insert 0=Disable 1=Enable	Number of Nulls To Send 000=1 Null 001=2 Nulls 010=3 Nulls 011=4 Nulls 100=5 Nulls 101=6 Nulls 110=7 Nulls 111=8 Nulls		
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 enables the interface to send out a series of ASCII null characters (NUL, character code 0) after end-of-line characters are sent on RS-232. (The end-of-line characters are specified by character registers C02 and C03.) This is used mainly by older-style printers that require a "fly back" time for the carriage to return. (The null characters don't take up space in the transmit buffer.)

Bits 2 through 0. Bits 2, 1, and 0 specify the number of nulls to send after the end-of-line characters.

Control Register R13

Control register R13 specifies which input signal lines the interface will monitor for initiating the autodisconnect sequence when that line becomes false. Refer to "Autodisconnect" on page 45.

R13 — Autodisconnect (Default 0000, Value=0)

Bit 3	Bit 2	Bit 1	Bit 0
Unused	Data Set Ready Disconnect 0=Disable 1=Enable	Received Line Signal Detect Disconnect 0=Disable 1=Enable	Clear To Send Disconnect 0=Disable 1=Enable
Value=8	Value=4	Value=2	Value=1

Bit 3. Bit 3 isn't used.

Bit 2. Bit 2 enables the interface to monitor the Data Set Ready line for a false condition and to disconnect the handshake when this line becomes false.

Bit 1. Bit 1 enables the interface to monitor the Received Line Signal Detect line for a false condition and to disconnect the handshake when this line becomes false.

Bit 0. Bit 0 enables the interface to monitor the Clear To Send line for a false condition and to disconnect the handshake when this line becomes false.

Character Registers

Character registers C00 through C11 specify the characters to be used for specifying end-of-line detect and delete characters, end-of-line insert characters, transmitter and receiver software handshake characters, RS-232 block sizes, and special characters. To define a character register, you need to store the appropriate character in that register. Refer to device-dependent messages and Remote mode instructions on pages 44 and 47 for details about setting these registers. For example, if you need to specify the four end-of-line characters, you can use the "LC" Remote instruction, or you can use the Device Dependent Listener 2 message followed by the appropriate data bytes that define these characters. Your controller may require that you use the binary values, the decimal values, or the characters as the proper input—refer to your controller's owner's manual. A table of the ASCII characters and their codes is presented in appendix E.

Character Registers C00 and C01

C00 — First End-Of-Line Detect/Delete Character (Default CR, Value=13)
C01 — Second End-Of-Line Detect/Delete Character (Default LF, Value=10)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies eight-bit code of character.							
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Character registers C00 and C01 store the values of the incoming RS-232 end-of-line characters that are to be detected and deleted. If there are two characters, then the first character goes in register C00 and the second goes in register C01. If there is only one character, then that character must go in register C01 and the value 0 must go in register C00. R10-3 must be equal to "1" to delete these characters. (If no characters are to be detected, R10-3 must be equal to "0".)

Character Registers C02 and C03

C02 — First End-Of-Line Insert Character (Default CR, Value=13)
C03 — Second End-Of-Line Insert Character (Default LF, Value=10)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies eight-bit code of character.							
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Character registers C02 and C03 store the values of the characters that are to be inserted as the end-of-line indicator. These characters may be used on RS-232, HP-IL, or both. If two characters are to be inserted, then the first character goes in register C02 and the second goes into register C03. If only one character is to be inserted, then that character must go in register C03 and the value 0 must go in register C02. R10-3 and R10-1 must be equal to "1" to insert these characters on HP-IL; R10-2 must be equal to "1" to insert these characters on RS-232. (If no characters are to be inserted on HP-IL or RS-232, R10-1 or R10-2 must be equal to "0", respectively.)

Character Registers C04 and C05

C04 — Receiver Protocol Ready Character (Default XON, Value=17)
C05 — Receiver Protocol Not Ready Character (Default XOFF, Value=19)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies eight-bit code of character.							
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Character registers C04 and C05 determine the characters that the interface uses for software handshake using receiver protocol. These characters are usually XON (DC1) and XOFF (DC3). These characters may be changed, but great care should be exercised to ensure that the characters used will be properly recognized by the RS-232 device. R11-3 must be equal to "1" to use receiver protocol. For additional detail on receiver protocol, refer to page 19.

Character Registers C06 and C07

C06 — Transmitter Protocol Request Character (Default ENQ, Value=5)
C07 — Transmitter Protocol Answer Character (Default ACK, Value=6)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies eight-bit code of character.							
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Character registers C06 and C07 determine the characters that the interface uses for software handshake using transmitter protocol. These characters are usually ENQ and ACK. These characters may be changed, but great care should be exercized to ensure that the characters used will be properly recognized by the RS-232 device. R11-2 must be equal to "1" to use transmitter protocol. For additional detail on transmitter protocol, refer to page 18.

Character Register C08

C08 — Transmitter Block Size (Default j, Value = 106)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies block size as described below.							
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

This register stores a binary number that is used to determine the size of the transmission blocks used with transmitter protocol (software handshake). The register value specifies the block size—the number of bytes to be passed in one sequence. The interface uses the actual value of C08 (except that 0 specifies 256 bytes).

If the interface is using transmitter protocol as a host device, it sends an ENQ after it sends the specified number of bytes and then waits for an ACK before sending the next block.

If the interface is using transmitter protocol as a terminal device, it will send an ACK in response to an ENQ only when at least the specified number of bytes are empty in the receive buffer. If the specified number is greater than 109, then the interface will never send an ACK.

> **Note:** With some controllers you may need to specify an ASCII character that has a numerical value equivalent to the number of bytes that are needed in a block. Refer to appendix E for the ASCII character set.

The ENQ and ACK characters mentioned above may be redefined by registers C06 and C07, respectively.

Character Register C09

C09 — Prompt Character (Default DC1, Value = 17)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies eight-bit code of character.							
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

This register defines the prompt character used with transmitter protocol when the interface is a terminal. To use this character, control register R11-2,1,0 must be equal to "101". Refer to page 19 for more information about using a prompt character with transmitter protocol.

Character Register C10

C10 — Receiver Block Size (Default CAN, Value = 24)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies block size as described below.							
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

This register stores a binary number that is used to determine the number of bytes reserved in the receive buffer for receiver protocol (software handshake). When the number of empty bytes decreases to the number specified by C10, the interface sends an XOFF. It sends an XON when the receive buffer becomes empty. The interface uses the actual value of C10 (except that 0 specifies 256 bytes). If the specified number is 109 or more, then the interface will never send an XOFF or an XON.

Note: With some controllers you may need to specify an ASCII character that has a numerical value equivalent to the number of bytes to be reserved. Refer to appendix E for the ASCII character set.

The XON and XOFF characters mentioned above may be redefined by registers C04 and C05, respectively.

Character Register C11

C11 — Delete Character (Default DC1, Value = 17)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Total value specifies eight-bit code of character.							
Value = 128	Value = 64	Value = 32	Value = 16	Value = 8	Value = 4	Value = 2	Value = 1

This register defines the selectable character that can be detected and deleted from RS-232 and not sent on HP-IL. To delete this character, control register R03-0 must be equal to "1".

Appendix E

ASCII Character Set

ASCII Char.	Character Code			
	Binary	Oct	Hex	Dec
NUL	00000000	000	00	0
SOH	00000001	001	01	1
STX	00000010	002	02	2
ETX	00000011	003	03	3
EOT	00000100	004	04	4
ENQ	00000101	005	05	5
ACK	00000110	006	06	6
BEL	00000111	007	07	7
BS	00001000	010	08	8
HT	00001001	011	09	9
LF	00001010	012	0A	10
VT	00001011	013	0B	11
FF	00001100	014	0C	12
CR	00001101	015	0D	13
SO	00001110	016	0E	14
SI	00001111	017	0F	15
DLE	00010000	020	10	16
DC1	00010001	021	11	17
DC2	00010010	022	12	18
DC3	00010011	023	13	19
DC4	00010100	024	14	20
NAK	00010101	025	15	21
SYN	00010110	026	16	22
ETB	00010111	027	17	23
CAN	00011000	030	18	24
EM	00011001	031	19	25
SUB	00011010	032	1A	26
ESC	00011011	033	1B	27
FS	00011100	034	1C	28
GS	00011101	035	1D	29
RS	00011110	036	1E	30
US	00011111	037	1F	31
space	00100000	040	20	32
!	00100001	041	21	33
”	00100010	042	22	34
#	00100011	043	23	35
$	00100100	044	24	36
%	00100101	045	25	37
&	00100110	046	26	38
’	00100111	047	27	39
(	00101000	050	28	40
)	00101001	051	29	41
*	00101010	052	2A	42
+	00101011	053	2B	43
,	00101100	054	2C	44
-	00101101	055	2D	45
.	00101110	056	2E	46
/	00101111	057	2F	47
0	00110000	060	30	48
1	00110001	061	31	49
2	00110010	062	32	50
3	00110011	063	33	51
4	00110100	064	34	52
5	00110101	065	35	53
6	00110110	066	36	54
7	00110111	067	37	55
8	00111000	070	38	56
9	00111001	071	39	57
:	00111010	072	3A	58
;	00111011	073	3B	59
<	00111100	074	3C	60
=	00111101	075	3D	61
>	00111110	076	3E	62
?	00111111	077	3F	63
@	01000000	100	40	64
A	01000001	101	41	65
B	01000010	102	42	66
C	01000011	103	43	67
D	01000100	104	44	68
E	01000101	105	45	69
F	01000110	106	46	70
G	01000111	107	47	71
H	01001000	110	48	72
I	01001001	111	49	73
J	01001010	112	4A	74
K	01001011	113	4B	75
L	01001100	114	4C	76
M	01001101	115	4D	77
N	01001110	116	4E	78
O	01001111	117	4F	79
P	01010000	120	50	80
Q	01010001	121	51	81
R	01010010	122	52	82
S	01010011	123	53	83
T	01010100	124	54	84
U	01010101	125	55	85
V	01010110	126	56	86
W	01010111	127	57	87
X	01011000	130	58	88
Y	01011001	131	59	89
Z	01011010	132	5A	90
[	01011011	133	5B	91
\	01011100	134	3C	92
]	01011101	135	5D	93
^	01011110	136	5E	94
_	01011111	137	5F	95
`	01100000	140	60	96
a	01100001	141	61	97
b	01100010	142	62	98
c	01100011	143	63	99
d	01100100	144	64	100
e	01100101	145	65	101
f	01100110	146	66	102
g	01100111	147	67	103
h	01101000	150	68	104
i	01101001	151	69	105
j	01101010	152	6A	106
k	01101011	153	6B	107
l	01101100	154	6C	108
m	01101101	155	6D	109
n	01101110	156	6E	110
o	01101111	157	6F	111
p	01110000	160	70	112
q	01110001	161	71	113
r	01110010	162	72	114
s	01110011	163	73	115
t	01110100	164	74	116
u	01110101	165	75	117
v	01110110	166	76	118
w	01110111	167	77	119
x	01111000	170	78	120
y	01111001	171	79	121
z	01111010	172	7A	122
{	01111011	173	7B	123
|	01111100	174	7C	124
}	01111101	175	7D	125
~	01111110	176	7E	126
DEL	01111111	177	7F	127

What To Do in Case of Difficulty

Summary of Potential RS-232 Problems

RS-232 is only partially defined as a standard. It allows variations in what lines are used and allows additional undefined lines to be added. Because of this, RS-232 can be troublesome even if two different manufacturers adhere to the standard. The standard specifically addresses Data Communication Equipment (modems) and Data Terminal Equipment. In situations involving only these two types of equipment, the interfacing difficulties are generally minimal. In the case of printers, which are not specifically addressed by the standard, compatibility problems are greater.

HP-IL/RS-232 Interface Summary

The HP 82164A HP-IL/RS-232-C Interface implements the RS-232 standard exactly as the standard specifies for Data Terminal Equipment using full-duplex protocol. Refer to appendix C for the signal lines that the interface implements.

The interface's features are:

- RS-232 Data Terminal Equipment and Data Communication Equipment pin and signal configurations.
- Full-duplex operation (with ability to operate with half-duplex devices).
- Hardware handshake options.
- Receiver (XON/XOFF) and transmitter (ENQ/ACK) software handshake protocols.
- Capability to insert or delete end-of-line characters.
- Fifteen different bit transmission rates.
- Four different word lengths.
- Manually controlled hardware handshake lines.

What If It Doesn't Work?

If you have difficulty getting your interface and external device to work together, follow this suggested list of things to check in troubleshooting the connection.

- Check the pin configuration and be certain which pin the device sends on and which it receives on. If it sends on pin 2, then it is a Data Terminal Equipment. If it sends on pin 3, then it is a Data Communication Equipment. Set the interface to the opposite configuration. (Refer to page 21.)
- Check the bit transmission rate (baud rate) on both the interface and the device—make sure they match.
- Check the number of bits the device is expecting to send and receive. This includes the start bit, bits in the data word, parity bit (optional), and stop bit(s).
- Next, check the handshake option—which signals the device is expecting to see true before it sends and which signals the device is expecting to see true before it receives. Remember that some devices expect to see Received Line Signal Detect (Carrier Detect) true before they will respond.
- Some devices require a software handshake. Check the software handshake option to be sure both devices are using the same protocol. Also check the block size and host/terminal option for transmitter protocol.

Using the HP-41 As a Controller

The HP-41 Handheld Computer, when used with an HP 82160A HP-IL Module, can interact with the HP-IL/RS-232 interface and its external device. However, the capabilities of this system aren't adequate to perform most typical operations involving the interface.

The HP 82183A Extended I/O Module extends the HP-IL capabilities of the HP-41 and HP-IL module and provides functions that are needed to control the interface and external device. This appendix contains information to help you use these products with your interface.

The table below lists functions that are useful for controlling your interface and external device. Refer to the owner's manuals for the HP-IL module and the extended I/O module for additional informaton about these functions and for information about other functions that may be useful in using your interface.

Function	Response
AID *	Sends one byte with the value 66.
CLRDEV *	Clears the buffers and resets the control and character registers to their default values.
CLRLOOP	Clears the buffers and resets the control and character registers to their default values.
DDL	If interface is a listener, performs the operation specified by the Device Dependent Listener message. (Refer to the table on page 28.)
DDT	If interface is a talker, performs the operation specified by the Device Dependent Talker message. (Refer to the table on page 28.)
DEVL †	Performs the operation specified by the Device Dependent Listener message. (Refer to the table on page 28.)
DEVT *	Performs the operation specified by the Device Dependent Talker message. (Refer to the table on page 28.)
FINDAID	If the X-register contains 66, places the address of the interface in the X-register.
FINDID	If the ALPHA register contains "HP82164", places the address of the interface in the X-register.
ID *	Sends the alpha string "HP82164A" (CR) (LF) to the ALPHA register. The CR LF aren't placed in ALPHA if flag 17 is clear.)
INA *	Retrieves data from the interface and stores it in ALPHA. Stops at CR LF (if flag 17 is clear) or at End Of Transmission message (when interface's receive buffer empties). (Leading bytes of value 0 aren't stored.)
INAC * INACL * INAE * INAN * INP * INXB *	Retrieves data from the interface, interprets it according to the particular function, and stores it in the HP-41. (For data stored in ALPHA, it is preceded by "D".)
IND *	Retrieves a sequence of bytes from the interface, then interprets the characters as a number and places it in the X-register.
INSTAT *	Retrieves the system status byte (byte 1) from the interface, sets flags 00 through 07 accordingly, and places number (modulo 64) in the X-register.
LAD	If the specified address matches the interface's address, makes the interface a listener. (If the address is 31, removes the interface from listener status.)
LOCAL †	Interface changes to Local mode, which sets it to transfer data to the external device. Doesn't prevent interface from changing to Remote mode when it next becomes a listener (if it has been enabled to do so, such as by REMOTE).
NOTREM	Disables interface from changing to Remote mode.

Function	Response
OUTA †	Sends all data bytes from ALPHA register to interface (followed by CR LF if flag 17 is clear). (Bytes of value 0 aren't sent.) In Local mode, the interface passes the data to the external device. In Remote mode, the interface interprets the data as Remote instructions.
OUTAC † OUTACL † OUTAE † OUTAN † OUTP † OUTXB †	Sends data bytes to the interface according to the particular function. In Local mode, the interface passes the data to the external device. In Remote mode, the interface interprets the data as Remote instructions. (For data from ALPHA, the first character isn't sent.)
POLL	Allows interface to indicate a service request condition (according to its parallel poll setup) and places parallel poll response in X-register.
POLLD *	Disables interface's parallel poll response.
POLLE *	Enables interface to respond to parallel poll (using POLL).
POLLUNC	Disables interface's parallel poll response.
REMOTE	If interface is primary device, changes interface to Remote mode, which sets it to interpret data as instructions. If interface isn't primary device, sets interface to change to Remote mode when it next becomes a listener.
SEND	Sends any HP-IL command message and causes the corresponding response. (Refer to the table on page 26.)
SRQ?	Allows interface to indicate a service request condition (according to its parallel poll setup) and branches according to whether any device requests service.
STAT *	Retrieves four bytes of status from interface and places the four corresponding characters in the ALPHA register (preceded by "S").
TAD	If the specified address matches the interface's address, makes the interface a talker. Otherwise, removes the interface from talker status.
UNL	Removes the interface from listener status.
UNT	Removes the interface from talker status.
XFER XFERC XFERCL XFERE XFERN	If interface is primary device, retrieves data from the interface and sends it to the specified device. If interface is specified device, sends data from primary device to the interface—in Local mode, interface passes data to external device; in Remote mode, interprets data as Remote instructions.

* The interface must be the primary device selected by the HP-41.

† The interface must be the primary device selected by the HP-41 or—under certain conditions—be a listener (as by using LISTEN or LAD).

The printer and general interface functions in the HP-IL module may also be useful for sending data to and from the interface and external device. To use these functions, ensure that the interface is the primary device selected by the HP-41 and that the calculator is in Manual mode (using MANIO).

Two examples of how you might use the HP-IL module and extended I/O module to control the interface are given below. These examples amplify the example on page 53 using the HP 2601A Daisywheel Printer. For this setup, the interface's internal jumper is used to configure the interface as a DCE.

The first example illustrates how you might use the Remote mode instructions to control your interface.

Assume that the connector is properly wired and that the printer is set for 1200 baud, even parity, receiver protocol software handshake, and full hardware handshake. The following program will set the control registers and prompt you for input to be printed. The end-of-line indicators are not automatically sent, so you need to send the CR and LF characters with your data. When you wish to stop, press CLA at the next prompt and then press R/S. Do not put more than 23 characters in the ALPHA registers or data will be lost.

01♦LBL "PRINT"	
02 AUTOIO	
03 "HP82164"	Specifies the interface's identity.
04 FINDID	Finds the interface's address.
05 SELECT	Selects the interface as the primary device.
06 REMOTE	Sets the interface in Remote mode.
07 "SB8"	Control register Remote mode instruction.
08 OUTA	Sends the Remote mode instruction to the interface, followed by CR LF (flag 17 clear).
09 "P0"	
10 OUTA	
11 "C2"	
12 OUTA	
13 "SL0"	
14 OUTA	
15 NOTREM	Returns the interface to Local mode.
16♦LBL 01	
17 "DATA?"	Prompt message.
18 AON	Activates Alpha mode.
19 PROMPT	Prompts for alpha input.
20 AOFF	
21 ALENGIO	
22 X=0?	Checks for no input
23 GTO 02	Branches for no input.
24 68	Decimal value of the ASCII "D" character.
25 XTOAL	Inserts the character "D" on the left of the ALPHA Register.
26 RDN	Returns the original length of the ALPHA register to the X-register.
27 OUTAX	Sends the contents of the ALPHA register except for the left-most character "D" to the interface.
28 GTO 01	
29♦LBL 02	
30 12	Decimal value of the ASCII form feed (FF) character.
31 OUTXB	Sends FF to the interface.
32 END	

The following program is the same as that in the previous example except that the control registers are set using a Device Dependent Listener 0 message.

01♦LBL "PRINT"	
02 AUTOIO	
03 "HP82164"	Specifies the interface's identity.
04 FINDID	Finds the interface's address.
05 SELECT	Selects the interface as the primary device.
06 0	
07 DEVT	Sends a Device Dependent Talker 0 message
08 14	
09 INAN	Places the values of the 14 control registers in the ALPHA register.
10 8	Sets R07 (ninth character in the ALPHA register) to a value of 8.
11 ENTER	
12 9	
13 YTOAX	

14 6	Sets R08 to a value of 6.
15 ENTER	
16 10	
17 YTOAX	
18 0	Sets R09 to a value of 0.
19 ENTER	
20 11	
21 YTOAX	
22 8	Sets R11 to a value of 8.
23 ENTER	
24 13	
25 YTOAX	
26 LISTEN	Makes interface a listener.
27 ADROFF	Disables automatic Unlisten message.
28 0	
29 DDL	Sends a Device Dependent Listener 0 message.
30 OUTAE	Sends the revised contents of the ALPHA register to the control registers.
31 ADRON	Restores automatic message sending.
32♦LBL 01	
33 "DATA?"	Prompt message.
34 AON	Activates Alpha mode.
35 PROMPT	Prompts for alpha input.
36 AOFF	
37 ALENGIO	
38 X=0?	Checks for no input.
39 GTO 02	Branches for no input.
40 68	Decimal value of the ASCII "D" character.
41 XTOAL	Inserts the character "D" on the left of the ALPHA register.
42 RDN	Returns the original length of the ALPHA register to the X-register.
43 OUTAN	Sends the contents of the ALPHA register except for the left-most character "D" to the interface.
44 GTO 01	
45♦LBL 02	
46 12	Decimal value of the ASCII form feed (FF) character.
47 OUTXB	Sends FF to the interface.
48 END	

The last example involves two HP-IL systems communicating with each other through two interfaces. One interface must be set up as a DCE; the other interface must be set up as a DTE. Each interface is controlled by an HP-41. Each HP-41 should be running the following program. Both interfaces are set to their default conditions.

01♦LBL "INTFCE"	
02 AUTOIO	
03 "HP82164"	
04 FINDID	Finds the interface's loop address.
05 SELECT	Selects the interface as the primary device.
06♦LBL 01	
07 STAT	Retrieves the interface's four status bytes.
08 3	
09 ATOXX	Puts the second status byte in X-register.
10 X<>FIO	Sets flags 00-07 according to bit pattern of second status byte.

11 FC? 00	Tests for transmit buffer not empty (bit 0).
12 GTO 02	Branches for transmit buffer not empty.
13 "DATA?"	Prompt message.
14 AON	
15 PROMPT	Prompts for input.
16 AOFF	
17 68	Decimal value of the ASCII character "D".
18 XTOAL	Inserts a "D" in the left-most position in the ALPHA register.
19 OUTAE	Sends the contents of the ALPHA register on HP-IL.
20♦LBL 02	
21 INSTAT	Retrieves the interface's system status byte.
22 34	
23 X≠Y?	Checks for no data in the receive buffer.
24 GTO 01	Branches for no data in buffer.
25♦LBL 03	
26 INAE	Places the data from the receive buffer into the ALPHA register.
27 AON	
28 PSE	
29 GTO 01	

The HP 82160A HP-IL Module contains two functions that may be useful in operating your interface in its Remote and Local modes. These functions are REMOTE and LOCAL. The REMOTE function sends the Remote Enable message and then Listen Address and Unlisten messages to the primary device. If the primary device has Remote mode capability, the device will then be in Remote mode.

The LOCAL function in the HP 82160A HP-IL Module makes the primary device a listener and sends it a Go To Local message. All devices on the loop that have Remote capability will still be Remote enabled. If the interface is the primary device or is a listener, and if it has received the Go To Local message, then the interface will be in Local mode. However, the next time the interface receives its Listen Address message, it will again go into Remote mode.

For most HP-IL functions, the HP-IL module sends a sequence of messages on the loop before it actually performs the specified function. In the process of sending these preliminary messages, the HP-IL module sends a Listen Address message to the primary device, followed by an Unlisten message. If you select the interface as the primary device and execute LOCAL, the interface will be in Local mode. However, the next time you perform any function with the interface as the selected device, the HP-IL module will temporarily make the interface a listener, putting the interface back into Remote mode. The only time you may be able to use the LOCAL function to put the interface into Local mode is in certain programming situations where there is no printer on the loop or the printer is not tracing the program operation.

For most situations, your interface cannot be controlled using just the HP-IL module. If you are using the extended I/O module, you should avoid using the LOCAL function; instead, you should use the NOTREM function to return to Local mode. (The HP-IL module normally sends these preliminary HP-IL messages even if the I/O module is installed—although the ADROFF function eliminates the preliminary messages for certain extended I/O operations.)

Refer to page 28 for additional information about Local and Remote modes.

Using the HP-75 As a Controller

The HP-75 Portable Computer can control the interface and its external device when the HP-75 is used with the *HP-75 I/O Utilities Solutions Book* (part number 00075-13013). This solutions book provides the LEX file HPILCMDS, which provides all of the HP-IL messages listed in the HP-IL table on page 26. The following list gives the HP-75 BASIC statements that enable you to send HP-IL messages. Refer to the response table for the interface's exact response to a particular message.

`ASSIGN IO`

Assigns the devices on the loop and allows you to give each device a unique specifier.

`CLEAR LOOP`

Sends a series of Interface Clear messages.

`ENTIO$(`'*device code*'`,`'*message list*'`)`

Retrieves up to 256 data bytes from a device on the loop.

The device code is the name assigned to the device by an `ASSIGN IO` statement. They are entered as '`:`*xy*'. If no specifier is used, enter the null string ''.

The message list is one or more of the HP-IL message specifiers listed in the table below.* If more than one specifier is used, they are separated by commas.

`OFF IO`

Disables normal loop operations, device specifiers not usable, `SENDIO` and `ENTIO$` are still usable if you address the loop and use address numbers in your messages.

`RESTORE IO`

Restores normal loop operations, device specifiers are usable.

`SENDIO`'*device code*'`,`'*message list*'`,`'*data list*'

Sends messages and data on the loop.

The device code is the name assigned to the device by an `ASSIGN IO` statement. They are entered as '`:`*xy*'. If no specifier is used, enter the null string ''.

The message list is one or more of the HP-IL message specifiers listed in the table below.* If more than one specifier is used, they are separated by commas.

The data list is the data characters you wish to send. If no characters are to be sent, then you must enter the null string ''. If you wish to enter character codes, you can use the `CHR$` function concatenated by the ampersand (`&`).

* For a detailed discussion about controlling peripherals using HP-IL messages, refer to *The HP-IL System: An Introductory Guide to the Hewlett-Packard Interface Loop* by Gerry Kane et al., Osborne/McGraw-Hill, Berkeley, California, 1982.

Message Specifiers

Specifier	HP-IL Message
AAD*n*	Auto Address 0–31. *n* is a number from 0 to 31 and is the starting address of the loop.
AAU	Auto Address Unconfigure.
AEP*n*	Auto Extended Primary 0–31. *n* is a number from 0 to 31 and is the extended address number.
AES*n*	Auto Extended Secondary 0–31. *n* is a number from 0 to 31 and is the starting address of the loop.
CL+	Inserts carriage return and line feed characters onto the end of a group of incoming data bytes.
DCL	Device Clear.
DDL*x*	Device Dependent Listener 0–31. *x* is a number from 0 to 31 and is the number of the Device Dependent Listener message to be sent.
DDT*x*	Device Dependent Talker 0–31. *x* is a number from 0 to 31 and is the number of the Device Dependent Talker message to be sent.
EAR	Enable Asynchronous Requests.
EON	Enable Listener Not Ready.
GTL	Go To Local.
IFC	Interface Clear.
LAD*n*	Listen Address 0–31. *n* is a number from 0 to 31 and specifies the address of the device that should be a listener.
LAD#	Makes the device in the device code a listener.
LPD	Loop Power Down.
NOP	No Operation.
NRD	Not Ready For Data.
NRE	Not Remote Enable.
PPD	Parallel Poll Disable.
PPE*n*	Parallel Poll Enable 0–31. *n* is a number from 0 to 15 and is the number of the Parallel Poll Enable message.
PPU	Parallel Poll Unconfigure.
REN	Remote Enable.
SAD*n*	Secondary Address 0–30. *n* is a number from 0 to 30 and is the number of the Secondary Address message to be sent.
TAD*n*	Talk Address 0–31. *n* is a number from 0 to 31 and is the address of the device that should be the talker.
TAD#	Makes the device in the device code the talker.
TL+	Does not allow the HP-75 to send out an Untalk and Unlisten message after the current message string is completed.
SAI	Send Accessory ID.
SDA	Send Data.
SDC	Selected Device Clear.
SDI	Send Device ID.
SST	Send Status.
UNL	Unlisten.
UNT	Untalk.

The HP-75 does not automatically send an end-of-line (CR LF) sequence when the SENDIO statement is used. If you need an end-of-line sequence, you need to send those characters with your string.

The ENTIO$ statement does not automatically send a Send Data message. The HP-75 does not look for the end-of-line characters CR and LF when this statement is used.

The two examples below illustrate how you might use the HP-IL capabilities of the HP-75 and the LEX file HPILCMDS. These examples amplify the example on page 53 using the HP 2601A Daisywheel Printer. For this setup, the interface's internal jumper is used to configure the interface as a DCE.

Assume that the connector is properly wired and that the printer is set for 1200 baud, even parity, receiver protocol software handshake, and full hardware handshake. The following program will set the control registers using Remote mode instructions, then prompt you for input. Enter the name of the file you wish to print. This example assumes that you have already performed an `ASSIGN IO` statement and have assigned the device code `':IN'` to the interface.

`10 RESTORE IO`	Makes sure the loop is properly assigned.
`20 PWIDTH 80`	Sets 80 characters as the length of a line.
`30 PRINTER IS ':IN'`	Assigns the interface as the loop printer.
`40 SENDIO ':IN','REN,LAD#','SB8;P0;C2;SL0'&CHR$(10)&CHR$(13)`	Puts the interface in Remote mode and sets the control registers.
`50 SENDIO '','NRE',''`	Returns the interface to Local mode.
`60 INPUT 'FILE NAME?';A$`	Prompts for the name of the file to be printed.
`70 PLIST A$`	Prints the file.
`80 PRINTER IS *`	Removes the interface as the loop printer.
`90 END`	

The following program is the same as that in the previous example, except that the control registers are set using a Device Dependent Listener 0 message.

`10 RESTORE IO`	Makes sure the loop is properly assigned.
`20 DIM B$[14]`	Sets `B$` to a length of 14 characters.
`30 PWIDTH 80`	Sets 80 characters as the length of a print line.
`40 PRINTER IS ':IN'`	Assigns the interface as the loop printer.
`50 B$=ENTIO$(':IN','TAD#,DDT0,SDA')`	Retrieves the values of the control registers.
`60 B$[8,8]=CHR$(8)`	Sets R07 to a value of 8.
`70 B$[9,9]=CHR$(6)`	Sets R08 to a value of 6.
`80 B$[10,10]=CHR$(0)`	Sets R09 to a value of 0.
`90 B$[12,12]=CHR$(8)`	Sets R11 to a value of 8.
`100 SENDIO ':IN','LAD#,DDL0',B$`	Writes the contents of `B$` to the control registers.
`110 INPUT 'FILE NAME?';A$`	Prompts for the name of the file to be printed.
`120 PLIST A$`	Prints the specified file.
`130 PRINTER IS *`	Removes the interface as the loop printer.
`140 END`	

The last example involves two HP-IL systems communicating with each other using two interfaces. One of the interfaces must be set up as a DCE; both interfaces use their default register contents. Each interface is controlled by an HP-75. Each HP-75 should be running the following program. The program assumes that the file is a text type data file with no lines longer than 80 characters.

`10 RESTORE IO`	Makes sure that the loop is properly assigned.
`20 DIM C$[80]`	Dimensions the string variable `C$` to 80 characters.
`30 INPUT 'FILE NAME TO SEND?';A$`	Prompts for file to send.
`40 ASSIGN #1 TO A$`	Opens the sending file.
`50 INPUT 'FILE NAME TO CREATE?';B$`	Prompts for the name of the file to be created. This file comes from the other HP-75.
`60 ASSIGN #2 TO B$,TEXT`	Opens the file to be created.
`70 ON ERROR GOTO 190`	Goes to line 190 when the end of the sending file is reached.

`80 C$=''`	Clears `C$`.
`90 D$=ENTIO$(':IN','TAD#,SST')`	Retrieves status bytes.
`100 E=NUM(D$[2,2])`	Selects second status byte.
`110 IF MOD(E,2)#1 THEN GOTO 90`	Checks for bit 0 (transmit buffer empty) equal to "1". This ensures that the other HP-75 is accepting the data.
`120 READ #1;C$`	Reads next data line into `C$`.
`130 SENDIO ':IN','LAD#',C$`	Sends `C$` to the interface.
`140 D$=ENTIO$(':IN','TAD#,SST')`	Retrieves the status bytes.
`150 E=NUM(D$[2,2])`	Selects the second status byte.
`160 IF MOD(IP(E/4),2)#1 THEN GOTO 100`	Checks for data in the receive buffer.
`170 PRINT #2; ENTIO$(':IN','TAD#,SDA')`	Writes the data from the receive buffer to the created file.
`180 GOTO 80`	
`190 FOR I=1 TO 5`	Begins a loop that checks for data in the receive buffer. If no data arrives after five loops, transmission is ended.
`200 D$=''`	
`210 D$=ENTIO$(':IN','TAD#,SST')`	Retrieves status bytes.
`220 E=NUM(D$[1,1])`	Selects first status byte.
`230 IF E#162 NEXT I ELSE GOTO 250`	If no data is in the receive buffer, then check again.
`240 GOTO 270`	Ends transmission after five loops.
`250 PRINT #2; ENTIO$(':IN','TAD#,SDA')`	If data is received, writes the data to the created file.
`260 GOTO 190`	Begins loop all over again.
`270 ASSIGN #1 TO *`	Closes output file.
`280 ASSIGN #2 TO *`	Closes created file.
`290 END`	

Index

Page numbers is **bold** type indicate primary references; page numbers in standard type indicate secondary references.

M

N

O

P

R

T

V

W

X

HP 82164A HP-IL/RS-232-C Interface
Owner's Manual

HEWLETT PACKARD

Corvallis Division
1000 N.E. Circle Blvd., Corvallis, OR 97330, U.S.A.

82164-90002 English

Printed in U.S.A. 4/83